Segmentation Strategies for Connectomics

A dissertation submitted by

Amelio Antonio Vázquez-Reina

in partial fulfillment of the

requirements for the Degree of

Doctor of Philosophy

in **Computer Science** at

TUFTS UNIVERSITY

Advisor: Eric L. Miller

2012

Approved by the Examining Committee:

- Eric L. Miller, Thesis Advisor
  Department of Computer Science, Tufts University.
  Department of Electrical and Computer Engineering, Tufts University.
- Hanspeter Pfister
  School of Engineering and Applied Sciences, Harvard University.
- Jeff Lichtman
  Department of Molecular and Cellular Biology, Harvard University.
- Lenore Cowen
  Department of Computer Science, Tufts University.
- Remco Chang
  Department of Computer Science, Tufts University.
- Kyongbum Lee
  Department of Chemical and Biological Engineering, Tufts University.

Tufts University
161 College Avenue
Medford, MA 02155

June 14th, 2012

Abstract

A human brain is estimated to have roughly 100 billion neurons connected through more than 100 thousand miles of axons and hundreds of trillions of synaptic connections. The full neural circuit within a brain is called its *connectome*, and understanding how it works and enables cognition, consciousness, or intelligence are important open questions in science. Recent developments in high-throughput electron microscopy imaging have enabled biologists to visually inspect brain tissue at resolutions of a few nanometers per voxel, enough to enable the analysis of neural circuits. However, the amount of data one would need to annotate to identify and reconstruct even a small circuit makes manual reconstruction efforts prohibitive.

In this thesis, we explore several computational strategies to facilitate the semi-automatic and automatic reconstruction of neurons from 3D stacks of connectomic images. We first propose Active Ribbons, a method based on deformable models and level set methods for tracing individual neurites that is amenable for interactive segmentation. We show that, unlike conventional level set methods, Active Ribbons can reliably capture neural membranes on electron microscopy stacks.

We then explore statistical models for automatic segmentation. We study the connection between the automatic segmentation of video and the reconstruction of connectomic stacks, and introduce Multiple Hypothesis Video Segmentation (MHVS), a method for the on-line segmentation of image sequences using long-term trajectories of 2D segments as possible labels. We demonstrate the applicability of MHVS in videos with an unknown number of objects and varying complexity. Building on the experience with MHVS, we propose Segmentation Fusion, a method for automatic neuron reconstruction that does not require the explicit discovery of labels a priori and that outperforms the state-of-the-art in automatic neuron reconstruction. We finally discuss several scaling strategies for distributed neuron reconstruction and show what we think are the largest neuron reconstruction results ever obtained in connectomics.

Dedicado a mis padres

Acknowledgments

I am deeply thankful for the support and guidance of my advisors. It has been an immense privilege and joy to work with Professors Eric Miller, Hanspeter Pfister and Jeff Lichtman. Their mentorship and advice enabled me as a scholar and as a researcher, and their example inspired me to pursue my Ph.D. and academic interests with the strongest commitment and passion.

I am very grateful to Sarah Frisken for inviting me to her lab meetings and for being the first one to believe in my potential as a Ph.D. student during my first year at Tufts. I very much enjoyed taking her graphics classes and participating in the graphics group. I am also very thankful to her for connecting me with Eric and Hanspeter.

I am thankful for Tufts University and the many professors, fellow students, and administrators who have supported me throughout my Ph.D. Tufts helped me believe in my academic and professional potential, first as an exchange student from Spain, and later as a graduate student. It has made me a better individual and profoundly enriched my life. I am proud to graduate from a school with a strong commitment to active citizenship and international outreach. Most importantly, Tufts introduced me to my wife (also a Jumbo), to new ways of thinking, and to a wonderful community.

I am thankful for Professors Lenore Cowen, Diane Suivaine and Judith Stafford for their encouragement and support during my first semester at Tufts. Their advice helped me navigate through a time of uncertainty and exploration.

I also want to thank my co-workers and collaborators; Steve Turney, Won-Ki Jeong, Mike Roberts, Ritwik Kumar, Verena Kaynig, Bjoern Andres, Bobby Kasthuri, Ed Soucy, Seymour Knowles-Barley, Daniel Huang, Michael Gelbart, Shai Avidan, and the rest of members of Hanspeter Pfister, Jeff Lichtman, and Eric Miller's groups. They provided me with invaluable feedback and insight. Working with all of them was a great pleasure and a constant source of inspiration.

I am profoundly grateful to my wife. Her unconditional love and support gave me infinite strength to enjoy every bit of my Ph.D. She was there for every up and down throughout this journey, providing me emotional comfort when I most needed it; and helping me keep work-life balance.

Lastly, and most importantly, I am deeply grateful to my family; and in particular to my parents. They made everything possible for me, and this thesis is dedicated to them. My dad was the first one to encourage me to pursue a Ph.D. and to come to Tufts. I owe him my passion for science and learning, and I know he left this world knowing that I was in the right place doing what I loved most.

Contents

Contents vii

1 Introduction 1

1.1 From Manual to Automatic Reconstruction 3

1.2 Image Segmentation and Connectomics 5

1.3 Connections with Video Segmentation 11

1.4 Interactive and Non-interactive Segmentation 12

1.5 Summary of Contributions 13

1.6 Thesis Organization 17

1.7 Publications Associated with the Thesis 18

2 Background and Related Work 21

2.1 Electron Microscopy Neuron Reconstruction 21

2.2 Variational Methods for Image Segmentation 25

2.3 From Active Contours to Level Set methods 26

The Level Set Equation 29

The Euler-Lagrange Equations 31
An Example: The Minimal Partition Problem for Image Segmentation 31
Level Set Methods and other Variational Segmentation Methods . . 34
2.4 Statistical Graphical Models 35
Markov Random Fields 36
Exponential Families 37
The Ising and Potts Models 38
Solving and Approximating MAP-MRF Problems 44
Combinatorial Optimization for MAP Inference 45
Message-passing Methods for MAP Inference 48

3 Semi-automatic Segmentation with Active Ribbons 51
3.1 Synopsis 51
3.2 Introduction 52
Contributions 54
Related work 55
3.3 Minimum Partition with Multiphase Level Sets 56
3.4 Adding Geometric Priors to the Multiphase Framework 59
3.5 Active Ribbons 62
Force Field for Ribbon Consistency 63
Force Field for Ribbon-cell Interaction 66
Interaction Between Ribbons 67
3.6 Experiments 69
Setting the parameters of the model 71

3.7 Summary 73

4 Automatic Segmentation of Image Sequences with MHVS 79

4.1 Synopsis 79

4.2 Background 82

Contributions 83

Related Work 84

4.3 An Overview of MHVS 86

4.4 Enumeration and Scoring of Hypotheses 87

4.5 Hypotheses Competition 91

4.6 Experimental Results 99

4.7 Summary and Discussion 104

5 Segmentation Fusion for Connectomics 107

5.1 Synopsis 107

5.2 Background 108

From MHVS to Fusion 109

Contributions 111

5.3 Segmentation Fusion for Neural Reconstruction 113

5.4 Modeling Fusion with MAP-MRF 115

5.5 Computing 2D Pre-segmentations with the Adapted Zernike Features 120

5.6 Experimental Results 125

5.7 Summary 129

6 Scaling Strategies and Enabling Branching in Segmentation Fusion 135

6.1 Scaling Segmentation Fusion 136
6.2 Coping with Splits and Mergers in Fusion 138
6.3 Preliminary Results 144
6.4 Discussion 149
Parallelization 156

7 Future Work and Conclusions **158**
7.1 Future Work 158
"Smart Fusion" Methods 158
Active Reconstruction 160
Distributed Segmentation Strategies 162
7.2 Conclusions 163

A Appendix **167**
A.1 Linear Programming Formulation for the Ising Model 167

Bibliography **170**

CHAPTER 1

Introduction

A human brain is estimated to have roughly 100 billion neurons connected through more than 100 thousand miles of axons and hundreds of trillions of synaptic connections ($\approx 10^{15}$ or 2^{50} connections) [1,2]. Given the size and complexity of the nervous system, it is not surprising that the neural circuits underlying even simple behaviors are not known.

The full neural circuit within the brain is called its *connectome*, and understanding how it works and enables cognition, consciousness, or intelligence are important questions in science [1,3–5]. The study of connectomes and patterns of neural connectivity has given rise to *connectomics*, an emerging area of neuroscience dedicated to the high-resolution and high-throughput reconstruction of nervous systems.

One of the early milestones in connectomics was the complete identification and reconstruction of the full nervous system of the worm *Caenorhabditis elegans*, most commonly known as *C. elegans*, in the 1970s and 1980s [6]. The approximately 7000 connections and 300 neurons of the *C. elegans* were mapped using Electron Microscopy (EM) over a period of ten years, with a significant

portion of this time spent in tracing the neurons manually through imaged sections (slices) of brain tissue.

Until recently, attempts to reconstruct larger circuits with electron microscopy from more complex organisms than the *C. elegans* were not seriously entertained. However, in the last few years, modern advances in the preparation, sectioning and imaging of brain tissue have enabled biologists to image neural connectivity in relatively large volumes of brain tissue in a highly automated manner [7,8]. Such techniques can image close to 1000 cubic microns of brain tissue (10^{-9} cubic millimeters) with electron microscopy at resolutions of approximately 5 nm per pixel, and with a separation of roughly 30 nm between scans (see Fig. 1.1).

In contrast to imaging techniques based on optical fluorescence microscopy where light diffraction limits spatial resolutions to approximately 0.2 micrometers, electron microscopy resolutions are high enough to capture intracellular organelles and capture synapses, enabling the possibility of circuit reconstruction. The prospect of imaging, analyzing and comparing large circuits has motivated a large interest in reconstructing the nervous systems of several organisms such as the fruit fly [9], the zebrafish [10], and the mouse brain [7,11].

An example of a section of brain tissue imaged with a technique known as serial-section Scanning Electron Microscopy (ssSEM) is shown in Fig. 1.2. The section comes from the S1 Primary somatosensory cortex from an adult mouse. A close-up of this section is shown in Fig. 1.3. Both examples as well as all other images of brain tissue included in this thesis are from the mouse brain and are

courtesy of Jeff Lichtman's Lab at Harvard University, unless otherwise stated.

1.1 From Manual to Automatic Reconstruction

Although in the last few years there has been much progress in automating the imaging of large 3D volumes of brain tissue with electron microscopy techniques, the reconstruction of neurons has traditionally required the manual annotation of the imaged data. Unfortunately, the amount of data that one would need to inspect and trace to reconstruct even a small neural circuit makes fully manual efforts prohibitive. For instance, the example shown in Fig. 1.2 is part of a 3D volume that is 1,850 sections deep (one image per section), with pixels and one terabyte in size, uncompressed. Still this volume only captures $\sim$50 cubic microns of brain tissue, which is approximately a billionth of a cubic millimeter.

A recent study estimated that a manual reconstruction of a cubic millimeter would require up to hundreds of thousands of person-years [12]. This estimation is consistent with empirical observations made in our lab, where it took approximately two months to two post doctoral researchers to manually reconstruct a volume that was 30 cubic microns in size. Therefore, enabling the automatic or semi-automatic 3D reconstruction of neural circuits from such stacks is critical for the future of connectomics and is the main motivation of the research presented in this thesis.

In the following sections we refer to a *neural process* as any of the physical projections or protrusions from a neuron, i.e., a "branch" or an "arm" stemming

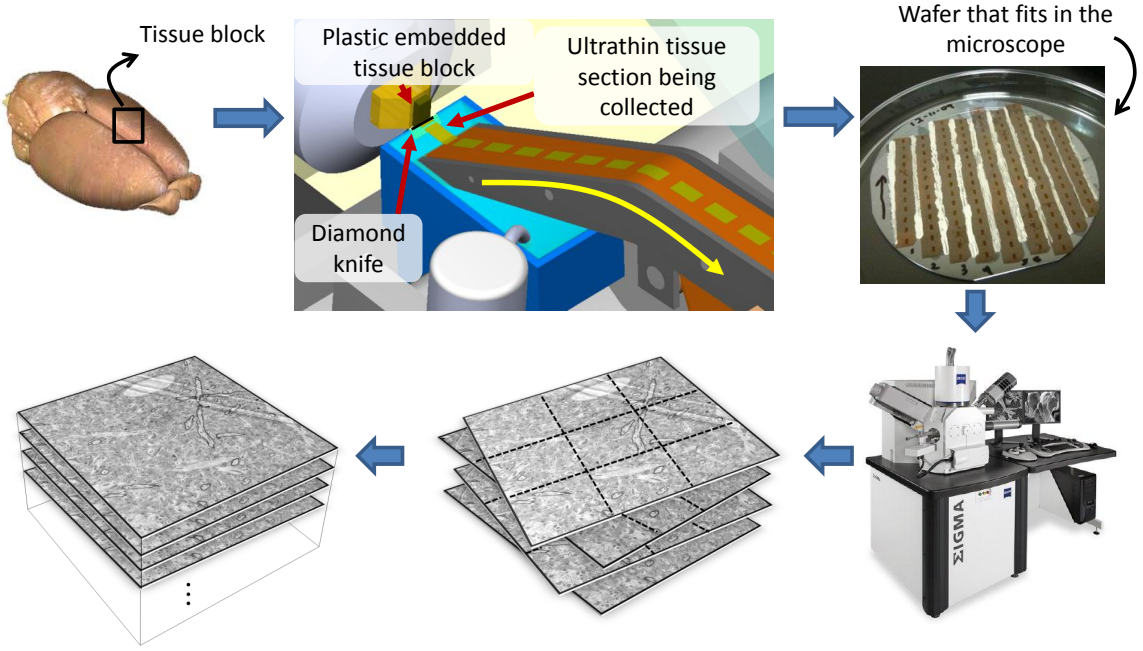


Figure 1.1: *With techniques like serial-section electron microscopy (ssSEM), biologists can image relatively large 3D volumes of brain tissue. To do this, the procedure is to first prepare the tissue and then cut it into ultra thin sections (∼30* nm*) that are collected in a tape. The sections are then placed in wafers to facilitate their sequential imaging under the electron microscope. The resulting images are then stitched and registered to form a 3D image stack. The 3D figure of the tape-collecting device above (known as ATUM) is courtesy of Kenneth Hayworth.*

from a neuron, such as a dendrite or an axon. We also refer to the reconstruction of a relatively small set of neural processes from a volume of brain tissue as a *sparse* reconstruction of the volume. Analogously, we refer to the segmentation of all neurons in a volume as a full or *saturated* segmentation of the volume.

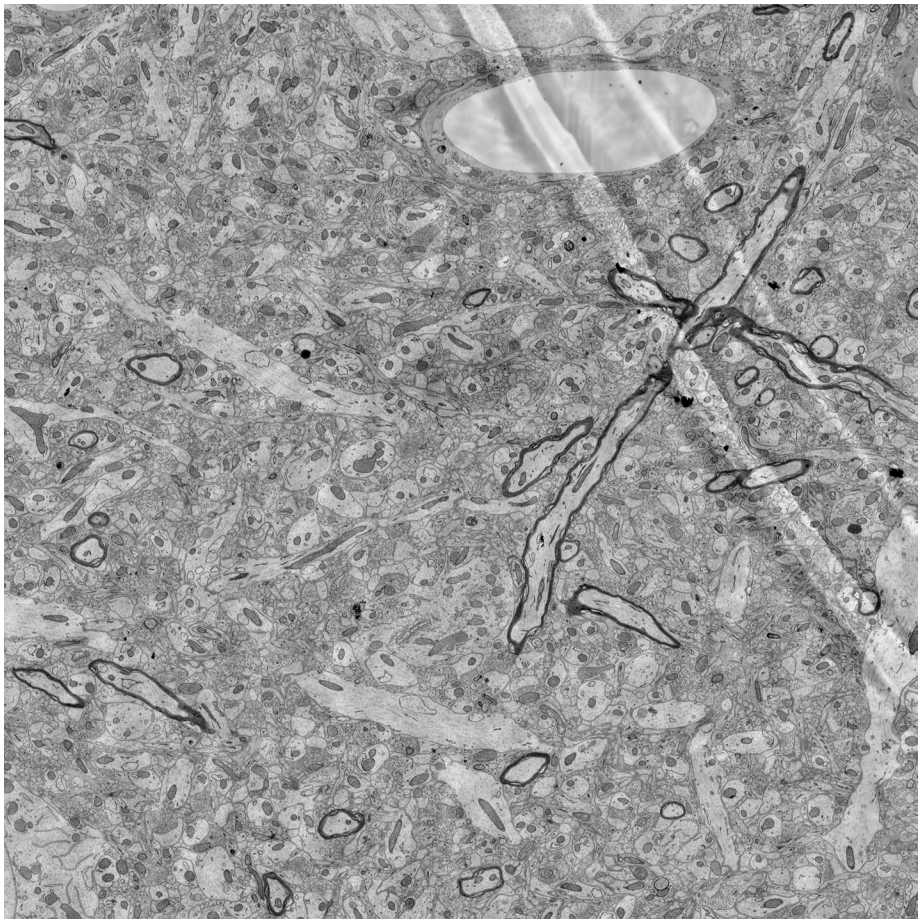


Figure 1.2: *Example from a section of mouse brain tissue imaged with electron microscopy. Each pixel is approximately* 5nm^2 *in size. The image a full resolution is approximately 10,000 x 10,000 pixels (the original section is ∼16K x 16K).*

1.2 Image Segmentation and Connectomics

The problem of reconstructing neural circuits from a connectomic stack of images is directly tied to the problem of obtaining the cell segmentation of the stack, namely the labeling of pixels in the stack according to which cell (neuron) the

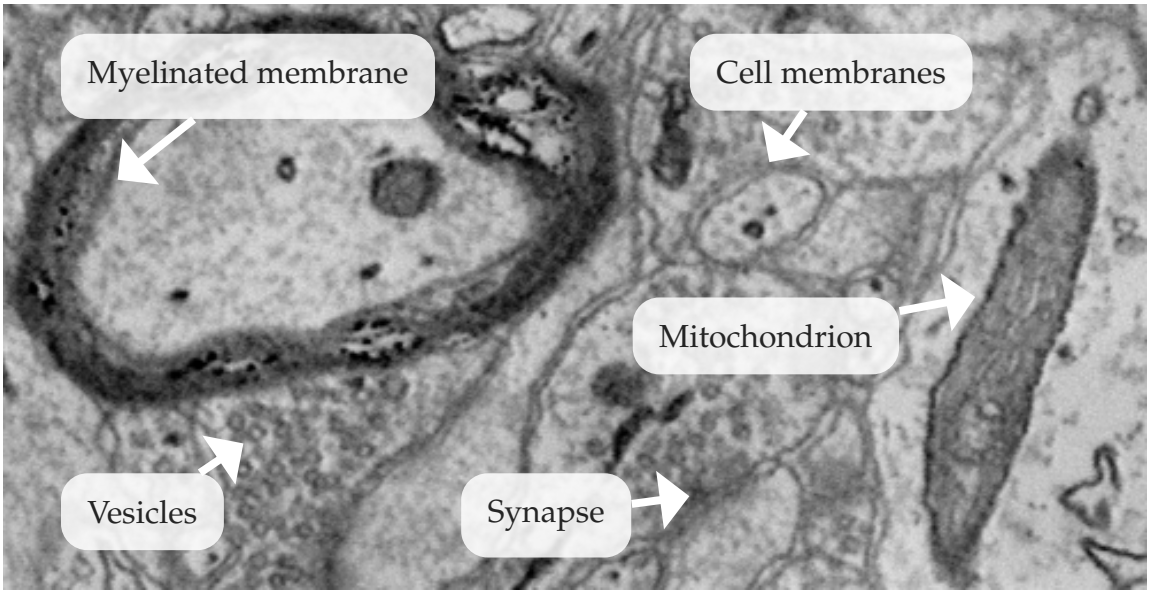
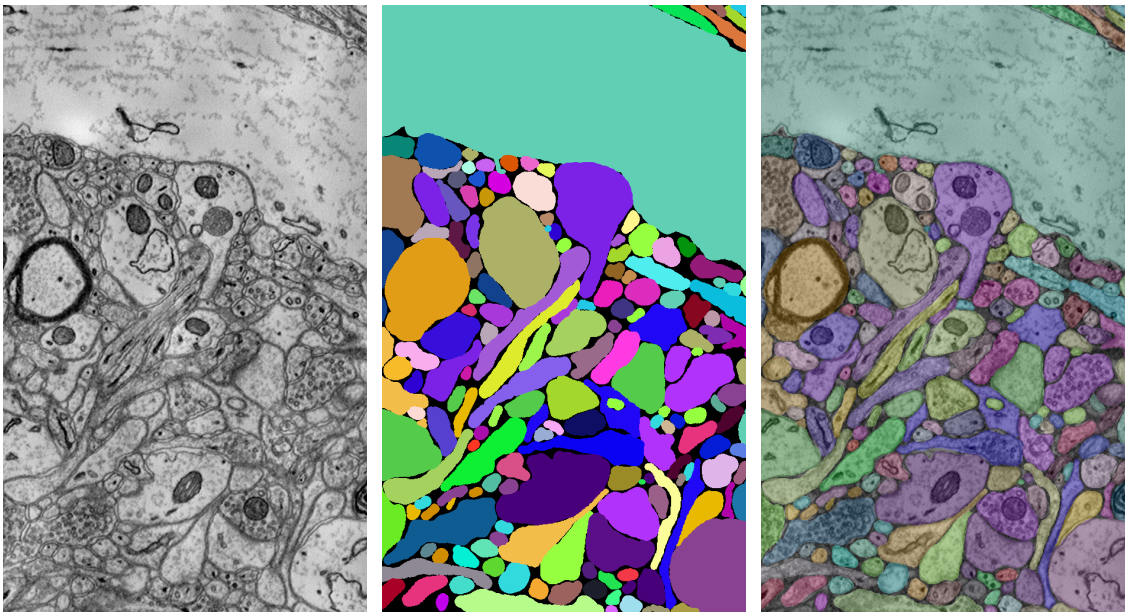


Figure 1.3: *A close-up of the image in Fig. 1.2. Electron microscopy techniques can image intracellular structures and allow for the visual identification of organelles such as mitochondria, neurotransmitters (vesicles), synapses, and myelinated axons. Each pixel is approximately* 5nm^2 *in size on the original image. Many synapses can be identified as regions where a thick dark cell membranes separates two neural processes, and a cluster of vesicles aggregates on one side of the membrane (on the pre-synaptic cell).*

pixels belong [13,14]. The segmentation of a 3D connectomic stack provides direct information about the location, trajectory and geometry of the neurons in the volume, and can facilitate the problem of identifying synapses between neurons in the image data [13]. Figure 1.2 shows an example of an example of manual segmentation of a small portion of a 2D section from the same 3D stack used in Fig. 1.2. As a comparison, Figure 1.5 shows an example of the automatic segmentation and reconstruction of a large section brain tissue obtained with one



(a) (b) (c)

Figure 1.4: *Example of a manual segmentation of neurons (courtesy of Daniel Berger). Each neuron is assigned to a different color. (a) Original image. (b) Labeling of the pixels in the image according to what neuron each pixel belongs (this is known as the "cell segmentation" of the image). (c) The label image superimposed on the original image (also known as the "stained-glass" visualization of the segmentation). The segmentation is supposed to be consistent across sections of the stack (i.e., the same neuron should get the same color in every section of the stack).*

of the automatic methods proposed in this thesis described in Chapter 5.

Automating image segmentation is a well discussed problem in the computer vision literature. A broad family of approaches to image segmentation involve integrating features such as brightness, color, or texture over local image patches and then clustering those features [15–19]. Other approaches are based on

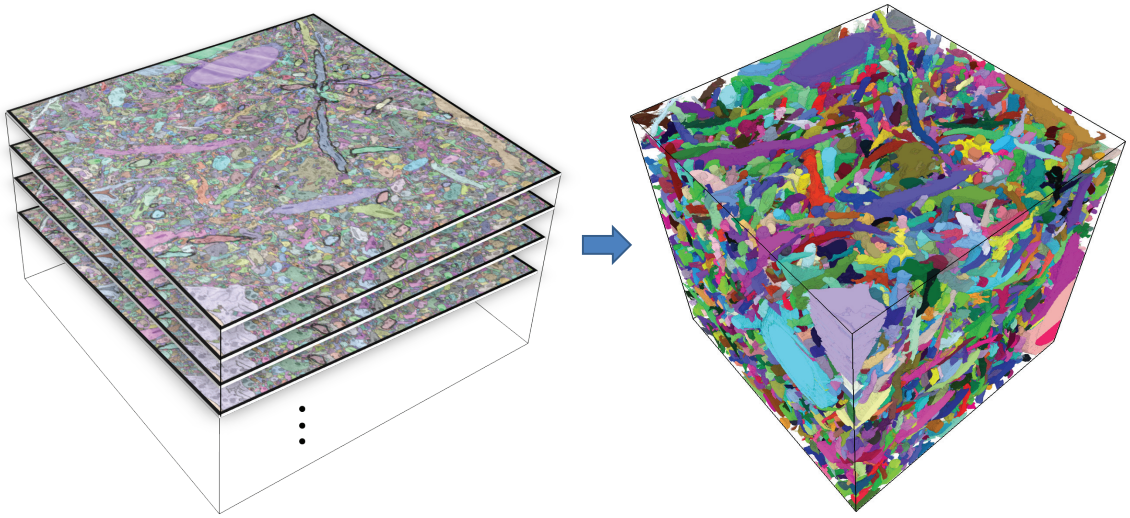


Figure 1.5: *The segmentation of the stack of ssSEM images shown in Fig. 1.1 using the automatic method described in Chapter 5. The original stack is 10,000 x 10,000 x 1000 images large. The sections were downsampled to 5,000 x 5,000 pixels for processing. The segmentation of the stack provides the 3D reconstruction of neurons.*

detecting edges or contours that can be used to partition the image into regions or segments [20–22].

However, although much progress has been done in image segmentation in the last few years, reliable automatic image segmentation is still an open problem in computer vision [21] and no automatic image segmentation is known to work well on all types of images. Moreover, automating the segmentation of connectomic stacks is particularly a challenging problem. Below we summarize some reasons [13, 23].

Unknown Number of Neural Processes and Topology

The number of neurons, neural processes and connections between neurons within a volume of brain tissue is generally unknown and difficult to estimate or bound. Neurons can branch, merge, originate or terminate anywhere within a volume, and some neural processes such as spines can be as thin as the distance between sections of electron microscopy stacks, or even move parallel to them, complicating their segmentation and tracing.

Image Quality and Changes in Appearance

Electron microscopy images are typically acquired with 8 bits of resolution per pixel in grayscale, and the contrast, focus, and signal to noise ratio can vary within and across sections of brain tissue. Additionally, the range of image intensity values of cell membranes overlaps substantially with that of other organelles. This complicates the detection of cell membranes using simple methods such as independently classifying each pixel according to its intensity alone; e.g., via image thresholding.

Size of the Images

The size of the images, hundreds of gigabytes per cubic micrometer of brain tissue, requires the design of computational segmentation methods that can scale. For example loading the entire dataset in memory at once is not a viable option. Still any method that attempts to segment the data by loading different

parts of the volume at any given time is supposed to a consistent segmentation (consistent labeling or coloring of all pixels) throughout the entire stack (i.e., pixels belonging to the same cell should be labeled consistently, not matter how large the cell is in the stack).

3D Anisotropy and Continuity across Sections

In techniques like serial-section Scanning Electron Microscopy (ssSEM), the tissue is first cut into thin sections that are automatically collected on a tape and later imaged independently under the electron microscope [24]. While cutting the tissue first and imaging it later allows for the scanning of relatively large volumes of brain tissue, methods like ssSEM depend on image registration to obtain a properly aligned 3D stack of images.

Technological difficulties in the cutting, collection and handling of thin sections of tissue limit the *off-plane* resolution (the resolution and separation between sections) to be ten times lower in practice than the 2D *in-plane* resolution that an electron microscope can achieve when scanning a section (typically 3-5 nm). The resulting stacks in ssSEM are thus said to be "anisotropic" in reference to difference in axis resolutions[1].

The anisotropy and potential errors in the registration of the volume result in stacks that show spatial discontinuities when moving between sections, even after image registration. These discontinuities can significantly complicate the

[1] Anisotropy typically refers to a property of being directionally dependent. In our case, the resolution of the 2D image that we see from the volume (as a projection) depends on the direction from which we look at the volume.

problem of visually tracking continuous structures across sections.

Segmentation Ambiguity and the Importance of 3D Context

Trained neurobiologists are sometimes unable to correctly discriminate cell membranes from debris or other cellular organelles on a given section without comparing it with consecutive sections. This may suggest that the image information on a single section may not be enough to properly identify physical cell membranes. Some experiments have also reported that neuroscientists can disagree on the 3D reconstruction even when considering all the available sections on a given stack [25].

1.3 Connections with Video Segmentation

A 3D stack of images of brain tissue can be seen as a sequence of 2D images, with consecutive images corresponding to consecutive sections in the volume of tissue. Neurobiologists often choose to inspect EM stacks by "flying" through the volume; a process similar to playing a movie where consecutive sections are assigned to consecutive frames. By visualizing sequences of EM sections at rates of 20-35 frames per second, neurobiologists can visually trace neural processes and get a sense of the trajectories neurons follow within a given stack.

Experiments like the one described above suggest an important connection between the problem of segmenting a 3D stack of images and that of segmenting videos. The relationship between video segmentation and connectomics inspired

and motivated some of the ideas and research presented in this thesis, in particular the work in the video segmentation presented in Chapter 4.

1.4 Interactive and Non-interactive Segmentation

Image segmentation methods can be classified according to various criteria, such as the assumptions they make about the data or the computational approach they follow (e.g., clustering of similar pixels, or detecting sharp edges on the image). Although several taxonomies are possible, in this thesis we pay particular attention at the difference between interactive and non-interactive segmentation methods.

Interactive or semi-automatic segmentation methods are those where the user helps the computer obtain a segmentation interactively providing help when needed. For example, the user may provide initial markers or seeds indicating how some pixels belonging to a neuron may be grouped. The user may also monitor the segmentation as it proceeds, correcting for errors or providing feedback when needed.

Interactive segmentation methods for connectomics are suitable for sparse reconstruction, for example when the user is willing to provide segmentation markers or seeds (e.g., by drawing scribbles or clicking inside cells), iteratively if necessary, to obtain the segmentation of a few neurons or neural processes in the stack.

Interactive segmentation methods can deliver relatively accurate

reconstructions, as the seeds and corrections provided by the user can help significantly reduce the ambiguity about the desired segmentation. In Chapter 3 we propose a method that is suitable for interactive segmentation known as *Active Ribbons*.

While interactive methods can help segment and reconstruct several neurons of interest from a stack, requiring the user to provide an initial seed or marker per neuron or per neural process is inviable for the complete or saturated segmentation of relatively large volumes. In comparison, non-interactive, or automatic segmentation methods are those where the user does not provide any initial estimates, nor corrects the segmentation as it proceeds. These methods may still be trainable via machine learning, but training is often provided *a priori* before segmenting a given volume of interest. We discuss and propose methods for the non-interactive segmentation of videos and EM stacks in Chapters 4 and 5.

1.5 Summary of Contributions

This thesis proposes novel segmentation methods for neuron reconstruction from electron microscopy stacks. The methods presented can effectively cope with challenges that are specific to high-throughput imaging techniques such as ssSEM, with anisotropic resolutions and large displacements of neural cross-sections between sections. Below we summarize the list of specific contributions:

1. Active Ribbons. We study deformable models that are suitable for interactive image segmentation and neural reconstruction. We show that conventional deformable models often fail to capture cellular membranes on electron microscopy images, missing important edges and incorrectly partitioning an image. To address this problem, we present a framework that allows for the direct modeling of elastic interactions between multiple level set functions. We then show how to use this framework to model Active Ribbons, a deformable ribbon-like model for image segmentation. The new level set formulation can capture ribbon-looking objects such as cell membranes of neural cross-sections more reliably. To the best of our knowledge, this is the first work that introduces a way of constraining the geometric arrangement of multiple level set functions on a given image.

2. Multiple Hypothesis Video Segmentation (MHVS). Motivated by the connection between connectomics and video segmentation that we described in Section 1.3, we present a novel solution to the problem of unsupervised *on-line* video segmentation.

In contrast to off-line video segmentation methods which may require random access to frames during processing, on-line methods process frames in a contiguous sequential manner, loading only a few frames at a time[2]. This type of processing is helpful when working with long sequences of images, such as deep connectomic stacks, where loading the entire stack in memory may not be feasible.

[2]On-line segmentation is different from *real-time* segmentation, which refers to the segmentation under hard time constraints.

MHVS is, to the best of our knowledge, the first method to introduce the notion of *deferred inference* to the problem of multi-label, on-line video segmentation. MHVS models segmentation as a competition of trajectories of regions of pixels within a sliding window of frames. The winning trajectories are determined as the *maximum a posteriori* solution to a Robust Potts Model. This model allows MHVS to segment arbitrarily long videos while encouraging label consistency between more than two frames.

3. Segmentation Fusion for Connectomics. An important component of MHVS is the enumeration of trajectories of pixel regions that can be selected as labels for segmentation within a window of frames. By enumerating a sufficiently large number of trajectories and letting the Robust Potts Model choose which ones to use to label pixels, MHVS can deal in practice with sequences where the number of labels is unknown *a priori*.

In an effort to propose a solution for unsupervised segmentation of stacks of images that avoids the explicit discovery of labels, we introduce Segmentation Fusion (or just Fusion). Instead of posing segmentation as the problem of directly identifying labels for pixels, in Fusion we formulate segmentation as the problem of choosing a set of segments and links between segments that together form a segmenting graph. The connected component labeling of this graph yields the final labeling (segmentation) of the stack of images.

4. Adapted Zernike Features. We make the observation that the detection of cell membranes as well as other cellular organelles on a given electron microscopy section of brain tissue should be invariant to rotations of the section. We then propose a rotationally-invariant set of features for pixel classification for connectomics. In contrast to other connectomic-specific descriptors from the literature, our features do not rely on a manually crafted set of filter banks, and are instead based on disk harmonics. We compare our features with competing descriptors and show they deliver state-of-the-art classification results.

5. Scaling segmentation methods for Connectomics. As part of ongoing work, we discuss two divide-and-conquer schemes for segmenting relatively large image stacks with Fusion. We propose the division of large connectomic volumes into a 3D grid of connectomic smaller subvolumes, where each subvolume can be segmented independently and in parallel using Fusion. We then evaluate two approaches for combining the subvolume segmentations into a final segmentation of the original full volume. We compare Bipartite Fusion, which matches the segmentation solutions for every pair of adjacent subvolumes independently, with *Nested Fusion*, an extension of Fusion that leverages multiple 3D pre-segmentations on each subvolume in an effort to integrate global context before reaching a final segmentation for the full volume. Finally we show what we think is the largest automatic neuron reconstruction result ever obtained with ssSEM.

1.6 Thesis Organization

The thesis is organized as follows. In Chapter 2 we discuss related work and introduce the mathematical foundations of level set methods and statistical graphical models. Level set methods provide the basis for Active Ribbons, while MVHS and Fusion are based on statistical graphical models.

We introduce framework for the direct modeling of elastic interactions between multiple level sets in Chapter 3. We use this framework to model Active Ribbons and show its applicability for interactive neuron segmentation.

We look into the problem of automatic segmentation of image sequences in Chapter 4. We begin by discussing previous work on video segmentation and introduce MHVS, a solution for the problem of multi-frame on-line video segmentation. We show examples of MHVS applied to natural videos with arbitrary camera and object motion.

In Chapter 5, we propose Segmentation Fusion and the Adapted Zernike Features for the problems of automatic multi-label segmentation of cellular organelles and the automatic cell segmentation of connectomic stacks. We compare Adapted Zernike Features with competing descriptors and evaluate Fusion against MHVS and other segmentation approaches based on agglomerative clustering and image co-clustering.

As part of ongoing work, in Chapter 6 we propose two strategies for scaling Segmentation Fusion and for obtaining the segmentation of large connectomic stacks. We also propose an extension of the original formulation of Segmentation Fusion that enables the method to cope with splits and mergers of neural

processes across sections of brain tissue.

We conclude in Chapter 7 where summarize the contributions of the thesis and give recommendations and possible directions for future research.

1.7 Publications Associated with the Thesis

The methods described in Chapters 3, 4 and 5 were published in the following three papers respectively:

[1] **A. Vázquez-Reina**, E. Miller, and H. Pfister, "Multiphase geometric couplings for the segmentation of neural processes," in *Proceedings of the 22nd International Conference on Computer Vision and Pattern Recognition (CVPR 2009)*.

[2] **A. Vázquez-Reina**, S. Avidan, H. Pfister, and E. Miller, "Multiple hypothesis video segmentation from superpixel flows," in *Proceedings of the 11th European Conference on Computer Vision (ECCV 2010)*.

[3] **A. Vázquez-Reina**, M. Gelbart, D. Huang, J. Lichtman, E. Miller, and H. Pfister, "Segmentation fusion for connectomics," in *Proceedings of the 13th International Conference on Computer Vision (ICCV 2011)*.

During the course of the research presented in this thesis, I also contributed to the following papers, some of which I discuss in subsequent chapters:

[1] W.-K. Jeong, J. Beyer, M. Hadwiger, **A. Vázquez-Reina**, H. Pfister, and R. T. Whitaker, "Scalable and interactive segmentation and visualization of neural

processes in em datasets," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, pp. 1505–1514, 2009.

[2] W.-K. Jeong, J. Beyer, M. Hadwiger, R. Blue, C. Law, **A. Vázquez-Reina**, C. Reid, J. Lichtman, and H. Pfister, "SSECRETT and NeuroTrace: Interactive visualization and analysis tools for large-scale neuroscience datasets," *IEEE Computer Graphics and Applications*, vol. 30, pp. 58–70, 2010.

[3] R. Kumar, **A. Vázquez-Reina**, and H. Pfister, "Radon-like features and their application to connectomics," in *Proceedings of the 23rd IEEE Conference in Computer Vision and Pattern Recognition Workshops (CVPRW 2010)*.

[4] M. Roberts, W.-K. Jeong, **A. Vázquez-Reina**, M. Unger, H. Bischof, J. Lichtman, and H. Pfister, "Neural process reconstruction from sparse user scribbles," in *Proceedings of the 14th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI 2011)*, 2011, pp. 621–628.

For reference, below we provide the list of authors who have contributed to some of the work presented in this thesis in alphabetical order:

- Charles Law
- Clay Reid
- Daniel Huang
- Eric Miller
- Hanspeter Pfister
- Jeff Lichtman
- Johanna Beyer
- Markus Hadwiger
- Michael Gelbart
- Mike Roberts

- Ritwik Kumar
- Ross Whitaker
- Rusty Blue
- Shai Avidan
- Verena Kaynig
- Won-Ki Jeong

CHAPTER 2

Background and Related Work

We begin with an overview of previous work for semi-automatic and automatic methods for neuron reconstruction. We then introduce the foundations for the methods proposed in this thesis and developed in subsequent chapters. We introduce two approaches for modeling and solving image segmentation; active contours and statistical graphical models. Active Ribbons (Chapter 3) are based on active contours, while MVHS (Chapter 4) and Fusion (Chapter 5) are based on statistical graphical models.

2.1 Electron Microscopy Neuron Reconstruction

The problem of the automatic analysis of ssEM datasets has been addressed by multiple authors in the last few years. These efforts have been devoted to a variety of problems, such as the labeling of cellular cross-sections [26, 27], the detection of specific cellular structures [28] such as mitochondria [29] or cell membranes [26, 27, 30], or the co-segmentation of pairs of adjacent sections of brain tissue [31].

A growing number of efforts have been dedicated recently to the problem of the automatic segmentation of neurons in electron microscopy images [26,27,30–32]. While some of these methods can cope with the 3D segmentation of anisotropic EM stacks [31,32], others only work on 2D [31], or rely on 6-wise pixel connectivity [30] for grouping pixels in 3D, making them unable to deal with anisotropic ssSEM stacks.

A common approach among segmentation methods for 3D anisotropic stacks is to first obtain a 2D segmentation of each section, and then cluster 2D segments across sections in the stack [26,31,32]. In contrast to the methods we present in Chapters 4 and 5, such clustering methods rely on the assumption that the initial 2D segmentation of each section is good enough for the posterior grouping, and that every neuron has been oversegmented in 2D [26,31]. Some of these methods, e.g., [26,32], also rely on greedy agglomerative clustering strategies and do not provide a measure of clustering optimality. Such methods require setting stopping conditions and heuristics to have the clustering of 2D segments in 3D converge to the right solution.

A number of efforts address the interactive and semi-automatic 3D sparse segmentation of individual neurons [33–35]. Jurrus et al. [33] use a deformable 2D contour on each section per neural process to trace the process through the 3D stack. The user provides an initial marker on the approximate center of a neural process on a given section, and the algorithm then segments and tracks the process section by section, iteratively sampling the image data and updating the segmentation from slide to slice using ideas from Kalman filtering theory.

Since the method uses a parametric contour that is discretized along its outline, the topology of the contour is fixed and supposed known a priori, and the contour must be re-discretized as it undergoes large changes in size and shape.

Pan et al. [35] propose a framework where the user provides 2D contours on two non-adjacent sections for each neural process. The 3D segmentation between both sections is then obtained from a set of 3D globally optimal paths that connect points between the 2D contours through the volume. Roberts et al. [34] propose a similar system where the user draws sparse scribble annotations on a neural process of interest instead of full 2D contours. The system then estimates 2D segmentations on these two sections using the scribbles as local constraints on a total variation minimization problem, applying the method of Unger et al. [36]. The 3D segmentation of the neural process between these sections is retrieved as the optimal volumetric pathway that connects these 2D segmentations.

Finally, a number of software tools and packages for neuron reconstruction have become available in the last few years. These tools vary in how much manual labor is required to obtain 3D neural reconstructions, and the type of user interaction employed.

One of the most widely used tools is Reconstruct[1] [37]. The tool has a simple 2D slice viewer with basic paintbrush editing functions to manually draw boundaries of neuronal cells. It has been used for several EM studies in the biology literature [38, 39].

[1]Reconstruct: http://www.synapses.clm.utexas.edu/tools/reconstruct/reconstruct.stm

A more complete tool than Reconstruct is TrakEM2[2], developed by Albert Cardona [40]. TrakEM2 provides a single package for data registration, navigation and segmentation. The tool integrates within Fiji[3], an image processing environment with hundreds of other image analysis plugins. The tool provides manual and some simple semi-automatic segmentation methods, such as region growing based on pixel similarity. The tool has been recently used in studies of the Drosophila brain [41]. A notable extension of TrakEM2 is CATMAID[4], a web-based front-end for TrakEM2 that allows for collaborative annotation on the Internet.

Another tool for neuron annotation and EM segmentation is Ilastik[5], introduced by Sommer et al. [42]. The tool allows for the interactive supervised training of a multi-label pixel classifier that can be used to detect cell membranes as well as other cellular structures in EM images. Ilastik also features a tool for 3D neuron reconstruction (i.e., cell segmentation) although it relies on 6-wise pixel neighborhoods to determine connectivity in 3D, and therefore is not directly applicable to anisotropic datasets.

Another recent tool for neuron reconstruction is KNOSSOS[6], which was recently introduced by Helmstaedter et al. [43]. KNOSSOS facilitates the manual reconstruction of cell skeletons, and can load datasets that may not fit in the computer's RAM. KNOSSOS has been recently used in studies of the detection of

[2]TrakEM2: http://www.ini.uzh.ch/~acardona/trakem2.html
[3]Fiji: http://fiji.sc/wiki/index.php/Fiji
[4]CATMAID: http://fly.mpi-cbg.de/~saalfeld/catmaid/
[5]Ilastik: http://ilastik.org/
[6]KNOSSOS: http://www.knossostool.org/

directed motion by the retina in mouse brain [44].

Finally our own lab at Harvard and Tufts has also written two software tools for the interactive neuron reconstruction. The first one is NeuroTrace [45] which is based on a 3D extension to Active Ribbons [46], the method we propose in Chapter 3. A second tool developed in our lab is Mojo, a GPU-based software suite for semi-automatic sparse segmentation based on the work of Roberts et al. [34].

For a more comprehensive comparison of these tools we refer the reader to the recent study of Helmstaedter and Mitra [23].

2.2 Variational Methods for Image Segmentation

In the last few decades the computer vision community has developed a large number of algorithms and models for the problem of image segmentation. A significant portion of these methods are *variational methods*, where segmentation is formulated as an optimization or mathematical programming problem [47]. In a variational method a quantity of interest measured by an objective function (e.g., a cost or energy functional) measures the quality of a segmentation, and the optimization of this objective gives rise to the desired solution. In contrast to ad-hoc segmentation methods based primarily on heuristics, variational methods facilitate the direct modeling of properties of desirable solutions and their quantitative comparison [48, 49].

Variational segmentation methods can be distinguished by the type of

objective function they use and by the techniques that can optimize them. Without loss of generality, we assume that an objective function is minimized in the optimization. A majority of variational methods for segmentation can be divided into two large groups [50]:

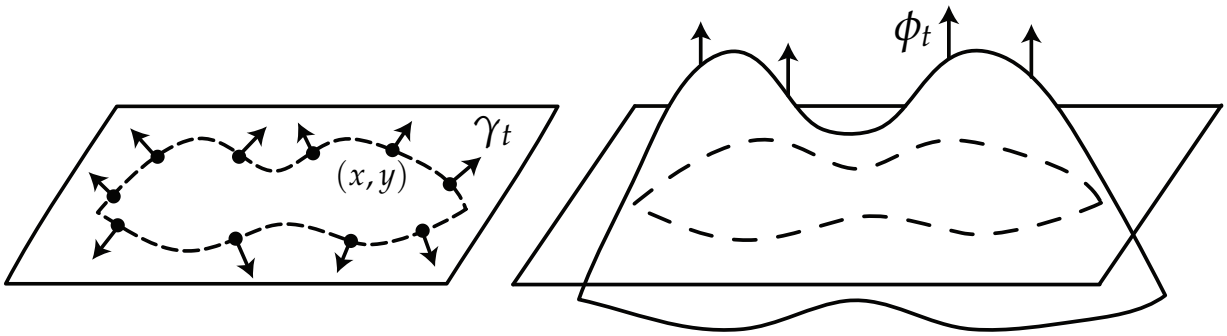
- Those requiring the optimization of an energy functional defined on a continuous grid, contour, or surface, e.g., [36,51,52].
- Those requiring the optimization of a cost function defined on a discrete grid or set of variables, e.g., [50,53,54]

Level set methods such as Active Ribbons, proposed in Chapter 3, fall in the first category, while variational inference methods based on statistical graphical models on discrete variables, such as MHVS and Segmentation Fusion (proposed in Chapters 4 and 5), fall in the second.

In the next section we introduce level set methods, a framework for representing and tracking deformable contours for segmentation. We then move onto statistical graphical models and discuss some examples of Markov Random Fields (MRF) for image segmentation.

2.3 From Active Contours to Level Set methods

The problem of 2D image segmentation can be formulated as the problem of partitioning an image into a set of regions or segments. One way to represent image regions is with closed contours.



(a) (b)

Figure 2.1: (a). *A parametric 2D active contour γ evolving with a velocity γ_t. The contour is discretized with a set of (x, y) markers. To track the evolution of the contour, the locations of the markers must be updated over time.* (b). *The same active contour encoded as the zero level set of a function ϕ. The evolution of the contour can be obtained by updating ϕ according to the velocity ϕ_t. At any time, we can find the location of the contour by finding the locations where $\phi = 0$.*

A well-studied approach to image segmentation is to use *active contours*. Active contours are deformable closed curves defined on the image domain that evolve from an initial shape and position (e.g., an initial guess provided by the user) towards the goal of partitioning an image according to some criteria. The evolution of active contours often follows the optimization of a quantity that typically measures the quality of the segmentation. This quantity is often referred to as the *energy* of the contour.

Some of the early active contours for image segmentation were represented with parametric geometric models. In one of the early popular active contour models known as *snakes* [55] an image segmenting contour is represented by a

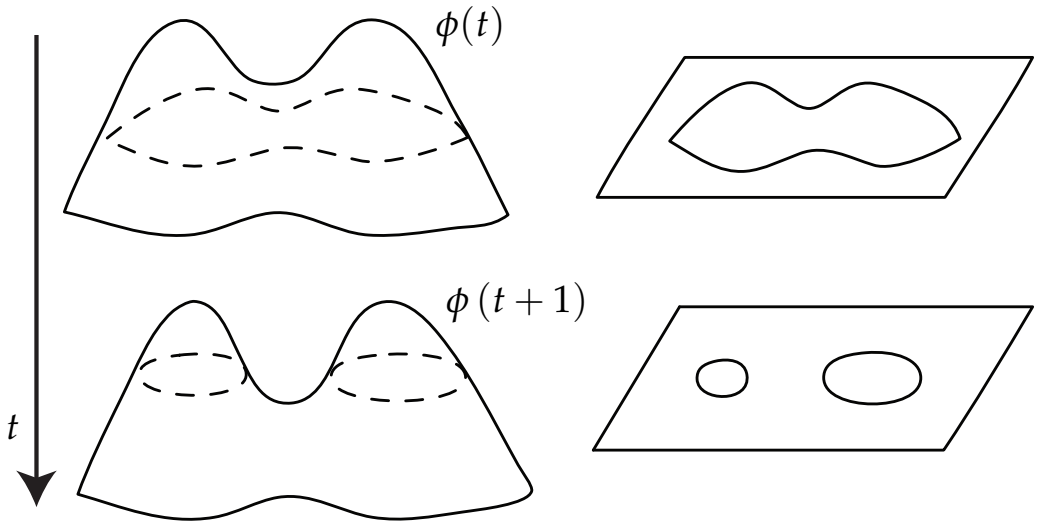


Figure 2.2: *Level set methods can automatically handle changes in the topology. For example, in this illustration a shrinking contour is made of only one component (top). As the contour shrinks, it may split into multiple components (bottom). The level set function ϕ can handle this change naturally. The level set representation avoids the need to keep track of markers in a discrete contour representation such as the one in Fig. 2.1(a). It also avoids the need to re-discretize the contour or detect when markers collide during the evolution.*

discrete set of points or (x, y) markers along the outline of the contour (e.g., see Fig. 2.1(a)). The optimization of the energy of the contour, for example via gradient descent, proceeds by iteratively updating the location of these points towards salient edges in the image until the optimization finishes. The final location of the points define the final contour that segments the image.

While parametric active contours such as those based on snakes enjoyed much popularity in the 1990s, the original models suffer from a number of limitations.

For example, models such as snakes have to be re-discretized (e.g., the markers need to be re-distributed) when the contour undergoes significant changes in size in order to keep the evolution of the contour accurate [56]. Perhaps most notably, parametric active contours are not capable of handling changes in the topology of the evolving shape unless special procedures, many times heuristics, are implemented for contour splits and mergers [52].

Level set methods provide an alternative mechanism for representing and tracking contours as they evolve towards segmenting an image. The segmenting contours are encoded in the zero level set (the zero-value iso-contour) of a function defined on a grid of the same dimensions as the image (see Fig. 2.1(b)). By embedding the segmenting contour in such function, the contour does not need to be re-discretized or re-parametrized as it evolves, and the contour can undergo changes in topology as the optimization proceeds (see Fig. 2.3). In practice, the flexibility of level sets to handle changes in topology is translated in the ability to segment objects in images where the topology or number of objects is not known *a priori*.

The Level Set Equation

The level set method allows to connect the shape evolution of a closed contour γ to the evolution of the zero level set of a function ϕ, so that the following equation holds over time:

$$\gamma(t) := \{(x,y) : \phi(x,y,t) = 0\}, \tag{2.1}$$

where (x,y) represent coordinates on the image domain, and t represents time.

In other words, the location of the contour γ at any point in time t is defined by the set of (x, y) locations at which $\phi(x, y, t)$ is 0 (see Fig. 2.1(b)).

This embedding couples the temporal evolution of ϕ with that of γ. Assuming that the contour γ moves with a velocity $\gamma_t = (x_t, y_t)$ and ϕ with a velocity of ϕ_t, we can obtain the relationship between these velocities by taking time derivatives on $\phi(x, y, t) = 0$, the equation encoding the zero-level iso-contour within the definition of Eq. 2.1. By the chain rule this results in:

$$\frac{\delta\phi}{\delta x}\frac{\delta x}{\delta t} + \frac{\delta\phi}{\delta y}\frac{\delta y}{\delta t} + \frac{\delta\phi}{\delta t}\frac{\delta t}{\delta t} = 0, \tag{2.2}$$

which can be rewritten as:

$$\nabla\phi \cdot \gamma_t + \phi_t = 0, \tag{2.3}$$

where $\nabla\phi = \left(\frac{\delta\phi}{\delta x}, \frac{\delta\phi}{\delta y}\right)$ is the gradient of ϕ on the image domain. The equation 2.3 is known as the *Level Set Equation* and it allows one to track the location of the contour γ as ϕ evolves over time.

Since velocities that are tangential to γ do not lead to changes in the shape of the contour [56], we only need the normal component of the velocity of the contour γ to track its shape evolution, and can write γ_t as $\gamma_t = \beta\,\hat{\mathbf{n}}$ where β is known as the contour speed and $\hat{\mathbf{n}}$ is the outward normal vector. Since γ is a level set of ϕ we also have $\frac{\nabla\phi}{|\nabla\phi|} = -\hat{\mathbf{n}}$ (assuming that ϕ is negative inside the contour, and positive outside) [57]. With this, we can rewrite the Level Set Equation as:

$$\frac{\delta\phi}{\delta t} = \beta|\nabla\phi|. \tag{2.4}$$

In practice, ϕ is often chosen to be the distance field of the contour γ on the image domain, which yields higher numerical stability [57] and satisfies the

property that $|\nabla\phi| = 1$, resulting in the equation:

$$\frac{\delta\phi}{\delta t} = \beta, \tag{2.5}$$

which can be updated with finite difference methods (see [56–58] for examples), and tracks the evolution of a contour γ that moves with normal velocity β.

The Euler-Lagrange Equations

When using active contours to iteratively solve a variational segmentation problem, for example, via gradient descent, one must know the conditions under which the active contour would make the original functional *stationary*, i.e., the conditions under which the gradient of the functional to be minimized is zero, defining a minimum of the function.

Such conditions can often be represented as solutions to partial differential equations known as the *Euler-Lagrange equations* of the original energy functional. The iterative solving of such Euler-Lagrange equations with finite difference methods [56, 58] gives rise to the optimization of the functional via gradient descent.

An Example: The Minimal Partition Problem for Image Segmentation

In this section we provide an example of an image segmentation problem on 2D gray scale images (images defined as scalar fields) whose solution can be approximated via gradient descent with the Level Set method. This example,

known as the *minimal partition problem* or the *Mumford and Shah piecewise-constant model* [59], serves as the basis for the introduction of Active Ribbons in Chapter 3.

In their seminal paper, Mumford and Shah proposed the idea to segment an image using a piecewise-constant approximation of the image. Roughly speaking they suggested the idea of segmenting an image by finding a "cartoon-like" approximation of the image defined by a set of 2D regions where the regions are "colored" with constant gray values that approximate the original image within each region. A well-known variation of this model is the *reduced model*, which for background-foreground segmentation can be stated as follows. Given a 2D image u, we seek a decomposition (i.e., a partitioning with no overlaps nor gaps) of the image domain Ω into two partitions Ω_1 and Ω_2, representing the foreground and background regions (i.e., the segmentation), and two c_1 and c_2 constants such that the following energy functional is minimized:

$$E(\Omega_1, \Omega_2, c_1, c_2) = \int_{\Omega_1} (u - c_1)^2 \, ds + \int_{\Omega_2} (u - c_2)^2 \, ds + \nu |C|, \tag{2.6}$$

where the first and second terms force the foreground and background regions to be similar to the constants c_1 and c_2, respectively, in the L_2 sense. The third term penalizes $|C|$, the length of the contour C that separates the 2D regions Ω_1 and Ω_2, encouraging C to be smooth (acting as a regularizing prior). The parameter ν balances the relative strength of the first two terms (also known as the data-fitting terms) and the contour smoothing term. Together Ω_1, Ω_2, c_1 and c_2 represent what's known as a "piecewise constant model of the image".

The functional above is known to not be convex [60,61], which complicates its optimization. In a series of papers, Chan and Vese [62–64] propose minimizing the functional above via gradient descent with the Level Set method. Using an indicator function to separate the regions Ω_1 and Ω_2 in the level set function ϕ, they modeled the minimization of Eq. 2.6 as the minimization of the following energy functional:

$$E(\phi, c_1, c_2) = \int_\Omega (u - c_1)^2 H(\phi)\, ds + \int_\Omega (u - c_2)^2 (1 - H(\phi)\, ds + \nu \int_\Omega |\nabla H(\phi)| ds, \tag{2.7}$$

where $H(\phi)$ is an indicator function known as the *Heaviside function*, defined as $H(\phi) = 1$ if $\phi \geq 0$, and $H(\phi) = 0$ otherwise. In the equation above, the image domain is represented as Ω, and Ω_1 and Ω_2 are defined as the regions where $\phi \geq 0$ and $\phi < 0$, respectively, as given by regions where the $H(\phi)$ and $1 - H(\phi)$ are non-zero in each integral.

Since the functional in Eq. 2.7 has three unknowns, ϕ, c_1 and c_2, Chan and Vese deduce three Euler-Lagrange equations to iteratively minimize the functional. Keeping the constants c_1 and c_2 fixed, they obtain the equations for ϕ:

$$\frac{\delta \phi}{\delta t} = \delta(\phi) \left[\nu \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - (u - c_1)^2 + (u - c_2)^2 \right], \tag{2.8}$$

and, proceeding similarly for c_1 and c_2, they obtain:

$$c_1 = \frac{\int_\Omega u H(\phi)}{\int_\Omega H(\phi)}, c_2 = \frac{\int_\Omega u (1 - H(\phi))}{\int_\Omega (1 - H(\phi))}. \tag{2.9}$$

Note that in the Equations 2.9 above the constants c_1 and c_2 take the mean gray value of the image (u) in the regions Ω_1 and Ω_2 respectively, as specified by the partitioning of the level set function according to the Heaviside function $H(\phi)$.

Given the dependency between the three Euler-Lagrange equations, Chan and Vese propose minimizing Eq. 2.7 using a two-step approach, where the optimization proceeds by alternating the updates of ϕ in one step and c_1 and c_2 in a second step. The equations for iteratively solving the optimization problem with finite difference methods can be found in the original papers [62, 63].

Level Set Methods and other Variational Segmentation Methods

In addition to the level set implementation for the variational problem described in the previous section, level set methods have been applied to energy functionals where the active contour attempts to explicitly capture edges on images (e.g., pixels with sharp gradients). Some examples include geodesic active contours [52], where the penalty on the length of the active contour is weighted by the image gradient (rewarding smooth contours that sit on image gradients), or flux-maximizing active contours [65], where the active contour aims at locations where the flux of the gradient vector field of the image across the contour is maximized.

In some formulations it is also common to add external velocity fields (also known as "force fields") to the evolving contour to push the contour towards global image features of interest and try to avoid undesirable local minima in the optimization [66–68]. In Chapter 3, we propose how to use such force fields to encourage specific geometric arrangements of the partitions of the segmentation, such as "ribbon-like" partitions.

As Zhu and Yuille noted in [69], the piecewise-constant Mumford-Shah

functional described in the previous section also has a probabilistic interpretation. If we model the set of pixel intensities u_1 and u_2 in each region Ω_1 and Ω_2 as samples from two Gaussian distributions $p_1\left(u_1|\mu_1,\sigma_1\right)$ and $p_2\left(u_2|\mu_2,\sigma_2\right)$, the active contour C and parameters c_1 and c_2 that minimize the functional can be seen as the *maximum a posteriori* (MAP) estimates of a statistical inference problem. Under this interpretation, the minimization of the original energy functional can be understood as the minimization of a negative likelihood that factorizes according to the partitioning of the image into the regions Ω_1 and Ω_2, while encouraging a *Minimum Description Length* (MDL) in the piecewise-constant approximation of the image [69].

The minimal partition problem and the probabilistic interpretation given by Zhu and Yuille are an example of a variety of active contour models where the segmentation is formulated as an inference problem with priors, for example, on the image statistics of each partition [70,71], or on the shape of the segmenting contours [51]. We refer the reader to [72] for a review of statistical approaches to level set methods for segmentation.

2.4 Statistical Graphical Models

As we pointed out in the last section, the problem of image segmentation can be formulated as a problem of statistical inference. Given the image data and prior knowledge about the problem, we infer the most probable or maximum *a posteriori* value of a set of variables. The value of these variables encode the

solution to our segmentation problem by encoding, for example, the connectivity of the pixels in the image or the labels the pixels take in the segmentation.

In this section we provide a brief introduction of a class of statistical models called *graphical models*. Graphical models are the foundation to two of the methods proposed in this thesis, MHVS (Chapter 4) and Segmentation Fusion (Chapter 5).

We focus on a type of graphical models known as *Markov Random Fields* (MRFs), also known as *undirected graphical models*. We discuss MRFs defined on sets of discrete variables. We then highlight their applicability to the problem of image segmentation with two examples, the *Ising* and *Potts* models. As we see in (Chapter 4), MHVS is based on a variation of the Potts model known as the *Robust Potts Model*.

Markov Random Fields

Markov random fields facilitate the graphical representation of the factorization of a probability distribution, i.e., the decomposition of the distribution as a product of functions. This has two advantages. First, they provide an intuitive visual description of the factorization of the distribution and dependencies between variables, and second, they facilitate the use of efficient inference algorithms that exploit the graphical representation of the problem [47,73,74].

More specifically, a Markov random field is any distribution that factorizes as:

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \psi_c(\mathbf{x}_c) \tag{2.10}$$

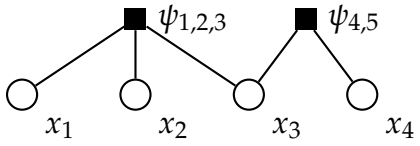


Figure 2.3: *Example of a factor graph for a distribution over four variables with two potentials. The factor graph specifies the factorization of the distribution as:* $p(x_1, x_2, x_3, x_4) = \frac{1}{Z}\psi_{1,2,3}(x_1, x_2, x_3)\,\psi_{3,4}(x_3, x_4)$. *The cliques* $\mathcal{C}$ *in Eq. 2.10 for this distribution (the sets of variables according to which the distribution factorizes) are* $\{1, 2, 3\}$ *and* $\{3, 4\}$.

where Z is a constant that ensures that $p(\mathbf{x})$ takes values between 0 and 1, and $\mathbf{x} = \{x_1, x_2, \ldots, x_m\}$ is a vector of random variables. The functions $\psi_c \geq 0$, also known as *potentials*, are defined on cliques (sets) $\mathbf{x}_c$ of random variables x_i from some set of cliques $\mathcal{C}$.

Markov random fields accept multiple graphical representations. One common way of depicting them is with *factor graphs*, where each potential ψ_C is represented with a dark box and each variable x_i with a white circle (see Fig. 2.3 for an example).

Exponential Families

An important class of Markov random fields are those where the potentials can be described by an exponential form of the type:

$$\psi(\mathbf{x}_c) = \exp\left\{\sum_{\alpha \in \alpha_c} \theta_\alpha \phi_\alpha(\mathbf{x}_c)\right\} \tag{2.11}$$

where α indexes a set of functions ϕ_α known as *sufficient statistics* in a set of indices α_c, and the parameters θ_α are known as *exponential parameters* [47,73,74]. All the Markov random fields we discuss in this thesis can be represented with potentials of the above form and therefore model distributions that are members of the exponential family.

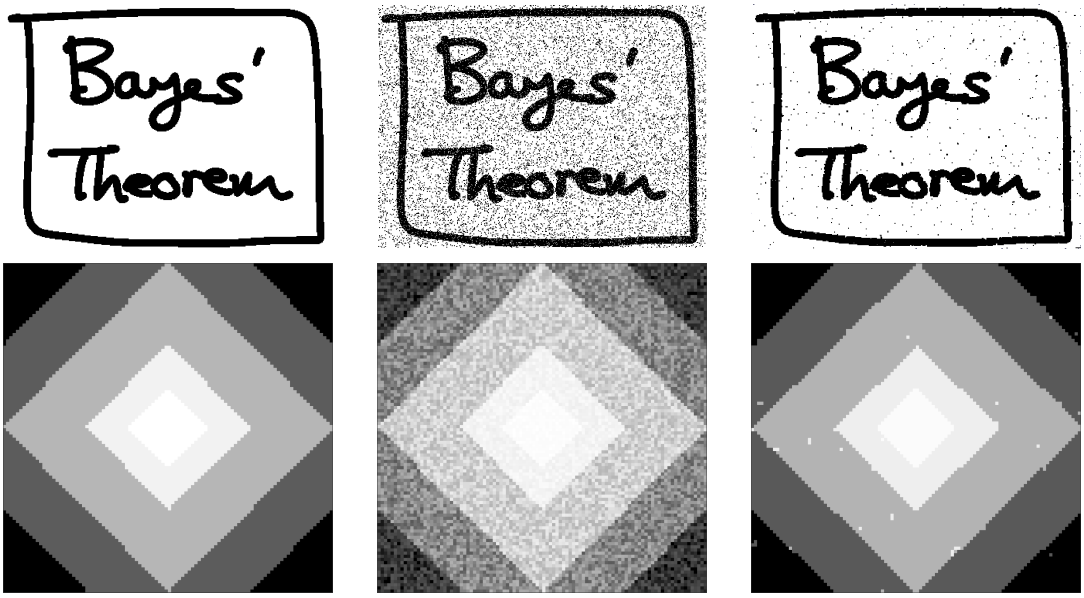
The Ising and Potts Models

Two examples of Markov Random Fields that have been used to model the problem of image segmentation are the Ising and Potts models [75–78].

To introduce these models we consider a related problem; restoring noisy images. As with segmentation, restoration can be understood as a problem of labeling pixels. In image segmentation, pixels are labeled with values that identify the object to which they belong (e.g., the neuron to which each pixel belongs), whereas in image restoration they are labeled with new intensity values; those that best restore the degraded image [79]. Under this interpretation, we can use the Ising and Potts models to model the restoration of, for example, binary images (with the Ising model) or grayscale images (with the Potts model). Fig. 2.4 shows examples of image restoration using these models.

In the following we introduce the Ising and Potts models using the problem of image restoration as an example. These models serve as the foundation of the Robust Potts model that we need for Chapter 4. We refer to [79] for more details about them.

As we mentioned at the beginning of this section, we can model image



(a) (b) (c)

Figure 2.4: *The restoration of noisy images can be understood as the process of labeling pixels with new intensity values, those that yield the best restoration of the image. We can model this problem as the maximum a posteriori solutions of Ising (top) and Potts (bottom) models for binary and grayscale image restoration, respectively.* (a) *Original images.* (b) *Images degraded with noise.* (c) *The result of the restoration using the methods of [80] (top) and [79] (bottom). The binary example was adapted from [81] and the grayscale example from [79].*

segmentation or restoration as the inference of the maximum *a posteriori* labeling of pixels given the image data (the pixel intensities on the original image).

We first define a random variable x_i for each pixel i on the image, representing the label (or new intensity value) that will restore the pixel. As we show next, if we assume that each such variable can take one of two possible values, 0 or 1,

e.g., for binary image restoration, we obtain the Ising model, while if we assume that the variables can take a higher number of values, (e.g., values from 0 to 255), one obtains the Potts model.

To formulate the problem image restoration, we model the probability associated with the desired restored image. Using the Ising model, we define this probability so that we are more likely to obtain a restored image that is similar intensity-wise to the noisy image. We also add to it the prior knowledge that neighboring pixels are in the restored image are more likely to have similar intensities. This prior knowledge can help remove the "salt-and-pepper" noise in the input degraded image [79]. To do this, we use two types of potentials, unary potentials acting on each each pixel, and pairwise potentials acting on neighboring pairs of pixels.

We thus define a probability distribution over all possible binary labelings (binary assignments to the unknown variables) that factorizes as a product of the unary potentials and pairwise potentials:

$$p\left(\mathbf{x}|\mathbf{I}\right) = \frac{1}{Z}\prod_{i\in\mathcal{V}}\psi_i\left(x_i\right)\prod_{(i,j)\in\mathcal{E}}\psi_{i,j}\left(x_i,x_j\right), \tag{2.12}$$

where $\mathbf{I}$ represents the image data, x_i represents the label (as a binary random variable) of a pixel i, $\mathcal{V}$ is the set of all m pixels on the image, and $\mathcal{E}$ represents the set of all neighboring pixel pairs on the pixel grid. The potentials $\psi_i\left(x_i\right)$ and $\psi_{i,j}\left(x_i,x_j\right)$ often also depend on the image data $\mathbf{I}$, but we drop this dependency for notational convenience.

The factor graph of this MRF for a 3x3 image is shown in Fig. 2.5.

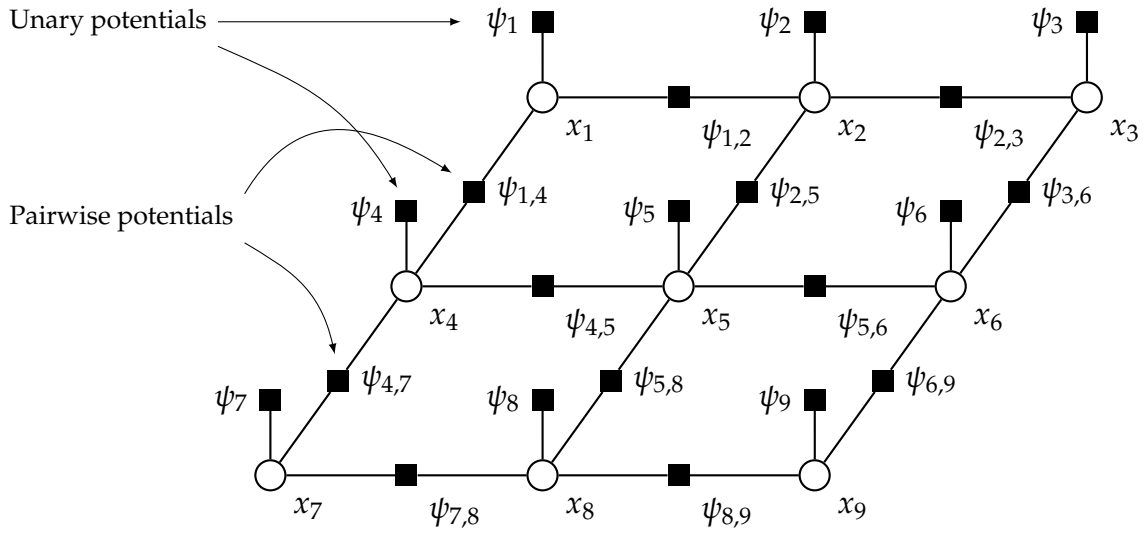


Figure 2.5: *Factor graph representation for the Ising and Potts models in Eq. 2.12 when defined on a 3x3 pixel grid on an image with 9 pixels. There is one random variable (white circle) for each pixel encoding the label (new intensity value) the pixel may take on the image. In the Ising model each variable can only take one of two values (e.g., 0 or 1 for binary image restoration), while in the Potts model each variable can take one of multiple values (e.g., from the range* $[0, 255]$ *for grayscale image restoration).*

We design the unary potentials ψ_i above so that darker pixels in the noisy image (with intensities closer to 0) are more likely to take the label 0 in the restored image, while lighter pixels (with intensities closer to 1) are more likely to take the new label 1. This can be done by setting:

$$\psi_i(x_i) = \exp\left\{-(x_i - I_i)^2\right\} \tag{2.13}$$

where I_i represents the image value on the pixel i. To see this, note that the more similar the label x_i of the pixel i is to its pixel intensity I_i in Eq. 2.13, the higher

the unary potential $\psi_i(x_i)$ is, and therefore the higher the probability in Eq. 2.12.

Similarly, we can design the pairwise potentials $\psi_{i,j}(x_i, x_j)$ so that neighboring pixels are more likely to take the same label, adding robustness to the restoration against noise. We can do this to add prior knowledge about what the nature of the restored image, encouraging a piece-wise constant labeling and restoration [77] of the noisy image, as we did with the Mumford Shah segmentation model described in Section 2.3.

More specifically, we can model the pairwise potentials $\psi_{i,j}$ as:

$$\psi_{i,j}(x_i, x_j) = \begin{cases} K & \text{if } x_i = x_j \\ 1 & \text{else} \end{cases} \tag{2.14}$$

where $K > 1$ is a constant that we can use to model the strength of this prior. Note that when neighboring pixels have the same label (i.e., $x_i = x_j$ for a pair of neighboring pixels on the pixel grid), the probability in Eq. 2.12 increases.

With all the potentials defined, we formulate the problem of image restoration as the maximum *a posteriori* labeling $\mathbf{x}^*$ of the MRF, i.e.,:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{I}), \tag{2.15}$$

where $p(\mathbf{x}|\mathbf{I})$ is given by Eq. 2.12. In the next section we discuss how solve and approximate solutions to this optimization problem. We show an example of the solution to 2.15 for restoring a degraded binary image (Ising model) and a grayscale image (Potts model) at the top of Fig 2.15.

Note that the probability distribution described with the potentials above is a member of the exponential family, as both the unary and pairwise potentials can

be written in the exponential form of Eq. 2.11. To see this, note that the pairwise potential $\psi_{i,j}\left(x_i, x_j\right)$ can be written as:

$$\begin{aligned}\psi_{i,j}\left(x_i, x_j\right) &= \exp\left\{\log_e\left(K\right) \cdot \mathbb{I}(x_i = x_j) - 0 \cdot \mathbb{I}(x_i \neq x_j)\right\} \\ &= \exp\left\{\log_e\left(K\right) \cdot \mathbb{I}(x_i = x_j)\right\} \qquad (2.16) \\ &= \exp\left\{\theta_K \cdot \mathbb{I}(x_i = x_j)\right\}, \qquad (2.17)\end{aligned}$$

where, following the notation in Eq. 2.11, θ_K is an exponential parameter, and the function $\mathbb{I}$ is a sufficient statistic known as the *indicator function* and is defined as: $\mathbb{I}(y) = 1$ if the condition y is true and $\mathbb{I}(y) = 0$ otherwise.

A similar expression can be derived for the unary potentials. If we assume that we are working with binary images and that the variables x_i take the values 0 or 1 (i.e., the Ising model), we can rewrite the potential $\psi_i\left(x_i\right)$ as:

$$\begin{aligned}\psi_i\left(x_i\right) &= \exp\left\{-\left(x_i - I_i\right)^2\right\} \\ &= \exp\left\{-(0 - I_i)^2 \cdot \mathbb{I}(x_i = 0) - (1 - I_i)^2 \cdot \mathbb{I}(x_i = 1)\right\} \\ &= \exp\left\{I_i^2 \cdot \mathbb{I}(x_i = 0) - (1 - I_i)^2 \cdot \mathbb{I}(x_i = 1)\right\} \\ &= \exp\left\{\theta_{i_0} \cdot \mathbb{I}(x_i = 0) + \theta_{i_1} \cdot \mathbb{I}(x_i = 1)\right\}, \qquad (2.18)\end{aligned}$$

where $\theta_{i_0} = I_i^2$ and $\theta_{i_1} = -(1 - I_i)^2$ are also exponential parameters. A similar expression can be derived for the Potts Model if we assume that the variables x_i can take a higher number of values, (e.g., for 8-bit grayscale restoration, values within the range 0 to 255):

$$\psi_i\left(x_i\right) = \exp \sum_{a=0}^{255} \theta_{i_a} \cdot \mathbb{I}(x_i = a), \qquad (2.19)$$

which is of the form given in 2.11, and where $\theta_{i_a} = -(a - I_i)^2$.

The Equations 2.18, 2.16 and 2.19 are said to re-parameterize the distribution of Eq. 2.12 in terms of a basis of sufficient statistics given by indicator functions [47].

Solving and Approximating MAP-MRF Problems

The problem of maximum a posteriori (MAP) estimation, i.e., computing the maximum a posteriori labeling, of a general Markov random field defined on a set of discrete variables is NP-hard in general, and a topic of active research. MAP estimation takes a special form when working with distributions of the exponential families. Using Eq. 2.10 and taking logarithms in 2.11 we have:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} p(\mathbf{x}) = \arg\max_{\mathbf{x}} \frac{1}{Z} \prod_{c \in \mathcal{C}} \psi_c(\mathbf{x}_c) = \arg\max_{\mathbf{x}} \sum_{\alpha \in \mathcal{I}} \theta_\alpha \phi_\alpha(\mathbf{x}) \tag{2.20}$$

where α indexes a set $\mathcal{I}$ of sufficient statistics $\phi_\alpha(\mathbf{x})$ of $p(\mathbf{x})$.

In the last few years, several approaches have been discussed for approximating or solving the problem in Eq. 2.20. These approaches can be roughly grouped into two categories, methods based on combinatorial optimization that rely on transformations to quadratic binary cost functions that can be solved later with graph cuts, and those that operate on linear integer programming and relaxations, such as belief propagation and message-passing methods. Determining which method is most appropriate for which type of MRF is an active topic of research, and we refer the reader to [76, 82, 83] for comparative studies between them. In the following we provide a brief overview of these two approaches, and show how they can be used to solve MAP inference problems such as the ones proposed in Chapters 4 and 5.

Combinatorial Optimization for MAP Inference

An important property of binary pairwise Markov random fields, i.e., MRFs with unary and pairwise potentials defined on binary variables, is that their MAP inference can be directly formulated as a Quadratic Psuedo-Boolean Optimization (QPBO) problem [74]. A QPBO problem is a problem where the cost function is quadratic on variables that are binary (i.e., boolean), and the cost function assigns real valued costs to each such binary input. Moreover, it is known that any MRF can be transformed into a binary pairwise MRF by adding auxiliary random variables to the original distribution without changing the marginal distribution of the original variables [47, 84, 85]. This is an important result since the MAP inference of any MRF can be formulated as a QPBO problem, and as we discuss next, such transformations enable the use of efficient combinatorial optimization algorithms (most notably graph cuts methods) for solving or approximating MAP inference.

As an example, consider the Ising problem of Eq. 2.15. Note that we can write this problem as:

$$\begin{aligned}\arg\max_{\mathbf{x}} p\left(\mathbf{x}|\mathbf{I}\right) &= \arg\max_{\mathbf{x}} \frac{1}{Z}\prod_{i\in\mathcal{V}}\psi_i\left(x_i\right)\prod_{(i,j)\in\mathcal{E}}\psi_{i,j}\left(x_i,x_j\right) \\ &= \arg\max_{\mathbf{x}} \sum_{i\in\mathcal{V}}\theta_K\cdot\mathbb{I}(x_i=x_j)+\sum_{i\in\mathcal{E}}\left(\theta_{i_0}\cdot\mathbb{I}(x_i=0)+\theta_{i_1}\cdot\mathbb{I}(x_i=1)\right).\end{aligned} \tag{2.21}$$

Since $\mathbb{I}(x_i = x_j)$ takes the same value as $x_i \cdot x_j$ (since both variables are binary), and similarly $\mathbb{I}(x_i = 1) = x_i$ and $\mathbb{I}(x_i = 0) = (1 - x_i)$, we can rewrite the

equation above as:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} \sum_{i \in \mathcal{V}} \theta_K \cdot x_i \cdot x_j + \sum_{i \in \mathcal{E}} \left(\theta_{i_1} x_i + \theta_{i_0} (1 - x_i) \right), \tag{2.22}$$

which involves the optimization of a quadratic cost function of binary variables.

While QPBO problems are known to be NP-hard in general, there are particular instances for which efficient algorithms are known. For example, MAP inference on Ising models with submodular (also known as regular) cost functions can be exactly solved in polynomial time [77]. Submodular cost functions are functions that exhibit the property of diminishing returns, namely functions f that satisfy $f(A \cup B) \leq f(A) + f(B) - f(A \cap B)$, where A and B represent sets of decisions in the optimization, i.e., the cost of combining two decisions in the optimization, such as activating two binary variables, is smaller than independently adding the costs of the individual decisions. For a binary pairwise MRF, sumbodularity is expressed as a function of the activation of individual variables in the MRF, and we say that the objective function in MAP estimation is submodular if for every pairwise potential we have $\psi_{i,j}(1,1) - \psi_{i,j}(0,0) \leq \psi_{i,j}(1,0) + \psi_{i,j}(0,1)$ [86]. The minimization of binary pairwise submodular function is equivalent to finding the minimum s-t cut in a graph [87], which can be solved using a maximum flow problem by the Ford-Fulkerson duality theorem [88]. Other QPBO problems such as the one corresponding to Ising models with negative pairwise potentials (also known as antiferromagnetic potentials), do not result in the optimization of a submodular cost function, and are known to be NP-complete [47]. For such non-submodular quadratic problems, methods based on roof duality such as the BHS

algorithm [89], or some of its recent extensions [90] provide an LP relaxation that can be solved with s-t cuts and give solutions with the property of persistency (partial optimality) [47].

When working with multi-label problems (i.e., MRFs with non-binary variables) such as the Potts model, there are known efficient approximation algorithms based on local search methods with transformations to QPBO problems that can also be solved with s-t cuts. Examples of such algorithms are the fusion-move [91], and the α-expansion and α-β swap move-making algorithms [77].

For higher-order problems (i.e., MAP of distributions with potentials involving more than two binary variables) several authors have also proposed transformations to QPBO problems, some of which can also be solved with graph cuts. Kolmogorov and Zabih [86] showed that all submodular functions of order three can be transformed to submodular functions of order two, and therefore can be exactly solved with s-t cuts. Freedman and Drineas [92] showed that certain submodular higher-order functions can be transformed to submodular second order functions by adding auxiliary binary variables. Ishikawa [84] showed that one can use similar variables to convert general, higher-order binary functions into quadratic functions. Finally, Rother et al. [85] extended these results to show that one can use similar reductions for general, higher-order multi-label MRFs.

An important aspect of the higher-order transformations mentioned above is that they may require an exponential number of auxiliary variables in the worst

case to convert a higher-order MAP problem into a second-order one [76]. However, for some specific classes of higher-order MRFs there are known transformations to quadratic problems where the number of auxiliary variables grows relatively slowly or remains constant. For example, in [93] Kohli et al. showed that for a higher-order variant of the Potts model known as the *Robust Potts model*, where large cliques of variables are encouraged to take similar labels, one only needs to add two auxiliary binary variables during the transformation to a quadratic problem. The Robust Potts model is the basis for the video segmentation method, MHVS, that we describe in Chapter 4.

In [94], Rother et al. made a similar observation by noting that for a wide class of binary higher-order MRFs in computer vision (e.g., such as those used for binary texture reconstruction), higher-order potentials are relatively sparse, i.e., they assign low costs to only a few configurations of the variables involved in the potential, and a fixed high cost to all other configurations. Such sparse higher-order potentials can be effectively approximated with products of pairwise potentials involving a small set of auxiliary variables. Rother et al. showed that these approximations can be used to transform sparse higher-order potentials into small products of binary pairwise potentials.

Linear Programming and Message-passing Methods for MAP inference

As we mentioned at the beginning of this section, a second important group of efficient methods for solving MAP-MRF problems are based on linear integer programming formulations and their relaxations. Examples of these methods

include belief propagation and other message-passing techniques [95–98]. The connection between message-passing methods and linear programming relaxations can be found in [47,74,98], where the authors show that message-passing algorithms can be understood as running block-coordinate descent on the Lagrangian dual of an LP relaxation.

Linear Programming (LP) refers to the optimization of linear functions subject to linear constraints, which is a well-studied problem in computer science and operations research [99]. In contrast to most of the combinatorial approximation methods based on graph cuts described in the previous section, MAP inference methods based on LP and its relaxations come with an optimal guarantee. If the LP relaxation is tight (i.e., the solution to the linear optimization problem is not fractional and satisfies integrality) then it is guaranteed to give the optimal solution to the original problem [100]. As an example of how to formulate a linear program for a MAP-MRF problem, in the Appendix A we provide an LP formulation of the Ising model that we discussed earlier in this Section.

While it is possible to use standard "off-the-shelf" integer programming or LP solvers that rely on methods such as the simplex method and branch-and-cut [99] to solve linear programs (as we do in Chapter 5), the graphical structure of the MRF can also be exploited to design distributed inference methods based on the notion of message-passing between nodes in the associated factor graph [47]. For sparsely-connected pairwise graphical problems (such as the Potts and Ising models), message-passing methods have been shown to be faster than standard LP solvers [74,83].

Examples of message-passing methods include belief propagation (BP) [101] or its loopy variant, LBP [95] and tree re-weighted variations of belief propagation such as TRW [98] or its convergent variant TRW-S [96].

The specific connection between these methods and the combinatorial optimization methods described in the previous section based on graph cuts is a topic of ongoing research. For example, in [102], Komodakis and Tzritas showed that the α-expansion combinatorial algorithm mentioned in the previous section can be seen as an iterative "primal integer-dual" algorithm for solving an LP relaxation.

All in all, level set methods and statistical graphical models provide powerful frameworks to formulate and solve variational inference problems in computer vision. Throughout this thesis we propose several methods based on these techniques for formulating and solving the segmentation of video and 3D connectomic stacks. In the next chapter, we start with the problem of detecting cell membranes in electron micrographs, and propose a method called Active Ribbons to facilitate their segmentation.

CHAPTER 3

Semi-automatic Segmentation with Active Ribbons

3.1 Synopsis

The ability to constrain the geometry of deformable models for image segmentation can be useful when information about the expected shape or positioning of the objects in a scene is known a priori. An example of this occurs in connectomics when segmenting neural cross sections in electron microscopy. Such images often contain multiple nested boundaries separating regions of homogeneous intensities. For these applications, level sets provide a partitioning framework that allows for the segmentation of multiple objects by combining several level set functions. Although there has been much effort in the study of statistical shape priors that can be used to constrain the geometry of each partition, none of these methods allow for the direct modeling of geometric arrangements of partitions.

In this chapter, we show how to define elastic couplings between multiple

level set functions to model ribbon-like partitions on an image. We build such couplings using dynamic force fields that can depend on the image content and relative location and shape of the level set functions. We call the resulting deformable models Active Ribbons.

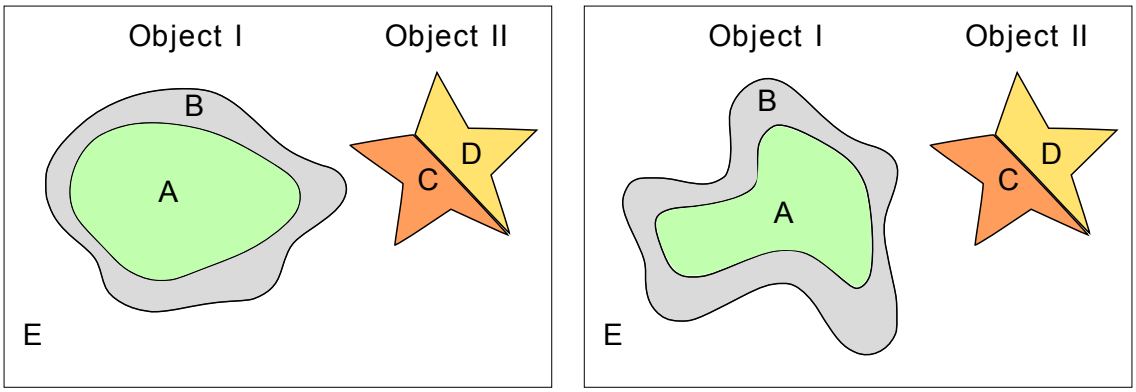
To the best of our knowledge, this is the first work that shows a direct way of geometrically constraining multiple level sets for image segmentation. We demonstrate the robustness of our method to capture cellular membranes by comparing it with other level set segmentation methods.

3.2 Introduction

As we mentioned in Section 2.3, deformable models based on level sets have been successfully applied to a variety of computer vision tasks such as image and video segmentation over the last ten years [103–106]. Their success is mostly attributed to their parametrization-free nature, intuitive formulation, and ability to easily adapt to shapes of unknown topology [107].

The problem of image segmentation (partitioning) within this framework is usually cast in a variational formulation; an energy functional is defined on the space of possible contours or image partitions (also known as phases), and the geometric deformable model is then iteratively evolved until an optimal solution is found.

A common approach to constrain the geometry of a deformable model is to build shape priors that are statistically learned from a set of training



(a) (b)

Figure 3.1: *Image segmentation of a scene with two objects. Subfigures (a) and (b) show a deformation of Object I. Previous level set methods for image segmentation do not allow for the addition of prior knowledge about the geometric arrangement of the partitioning. However, in this example we might know* a priori *that region B should always surround region A, and that the partitions C, and D form a different object that should not ever be surrounded by B. In this chapter we introduce a way of grouping and inducing such geometrical arrangements in the partitioning.*

templates [108–110]. However, such priors are limited to the subspace of learned deformations from the training set (typically up to an affine or a projective transformation), and they cannot directly model sophisticated geometric arrangements of deformable models. Such ability can be important in imaging applications, such as connectomics. For example, the segmentation of cellular and intracellular membranes in serial-section electron microscopy (ssSEM) images often requires partitioning the image into multiple nested contours. Such prior information about the image geometry can be used to avoid undesired segmentations and improve the overall segmentation accuracy.

In this chapter, we introduce a way to directly model geometric objects that are naturally described using multiple level set functions. As opposed to most of the previous work on multi-shape learning for deformable models [111,112], our method does not rely on the statistical inference of a multi-shape distribution from a set of training samples. We instead provide a way to directly design entire families of partition arrangements using dynamic force fields that can depend on the image content and multiple level set functions.

There are a number of important application areas where constraining the geometric arrangement of the partitioning is useful. Consider the illustration of Fig. 3.1 that shows a scene with two objects and the background. A classical multiphase method for segmentation would partition the scene into several phases according to image features (e.g., image intensity, gradient, etc.), and optionally, according to some statistically learned shape prior for each of the partitions. Such approaches do not naturally allow for constraining the relative arrangement of the regions in the partitioning. A real example of this idea can be seen in connectomics when tracking neural processes (cellular and intra-cellular boundaries) in a sequence of sections from a 3D volume of brain tissue. In each section, cells, mitochondria and other intra-cellular objects have a membrane of homogeneous intensity and varying thickness (see an example in Fig. 3.2).

Contributions

The main contributions of the work presented in this chapter are, first, the definition of elastic couplings between level set functions using dynamic "force

fields" to induce a geometric arrangement of partitions for image segmentation. And second, the modeling of ribbon-like deformable models that can be used to segment and track neural processes. To the best of our knowledge none of these issues have been addressed to date in the literature.

Related work

Variational and energy minimization models for image segmentation based either on level sets or graph cuts are well documented in the literature [50, 103, 113]. For multi-object segmentation, multiway graph cuts and multiphase level sets (i.e., segmentation with multiple level set functions) provide a natural extension of the single object case [60, 113–115].

In the multiphase level sets literature, much attention has gone into the study of topologically constrained flows that avoid vacuum regions and overlap between the different partitions [104, 116–118], and into extending shape priors for single level set evolutions to the multiphase case [112, 119–121]. Two interesting problems have been addressed in this last area, first, identifying which shape prior should be applied to which level set function or partition (this process is also known as shape prior competition) [119, 120], and second, applying shape priors while allowing for mergers and splits of multiple phases [121].

Recently, there has been an effort to employ external force fields within the multiphase level set framework in a manner which preserves the underlying, assumed known, topology of the problem [105]. The work presented in this

chapter shares some similarities with this work. Mostly, we both consider the integration of external force fields in multiple level set functions and their application in segmentation. However, our work focuses on how to induce a geometrical arrangement on the different phases or partitions, while the work of [105] focuses on preserving the topology of the different phases and avoiding vacuum regions and overlap altogether.

3.3 Minimum Partition with Multiphase Level Sets

To motivate our work, we revisit the two dimensional grayscale piecewise-constant segmentation problem in computer vision known as the *minimal partition problem* or the *Mumford and Shah problem* that was introduced in Chapter 2. As we recall from that chapter, this problem was originally formulated in [59], and since then, several variations have appeared in the literature.

In the following we introduce the generalization of the model discussed in Section 2.3 for m partitions. Let $\Omega \in \mathbb{R}^2$ be open and bounded, then, given an observed image u, we seek a decomposition $G = (\Omega_1, \ldots, \Omega_m)$ of Ω and a vector of constants $c = (c_1, \ldots, c_m)$ such that the following energy is minimized:

$$E(G, c) = \sum_{i=1}^{m} \left(\lambda_i \int_{\Omega_i} (u - c_i)^2 \, dx + \frac{\alpha_i}{2} \int_{\Gamma_i} ds \right), \tag{3.1}$$

where Γ_i represents the boundary set of each partition Ω_i and the tuning parameters $\lambda_i \geq 0$ and $\alpha_i \geq 0$ weigh the relative importance of the different

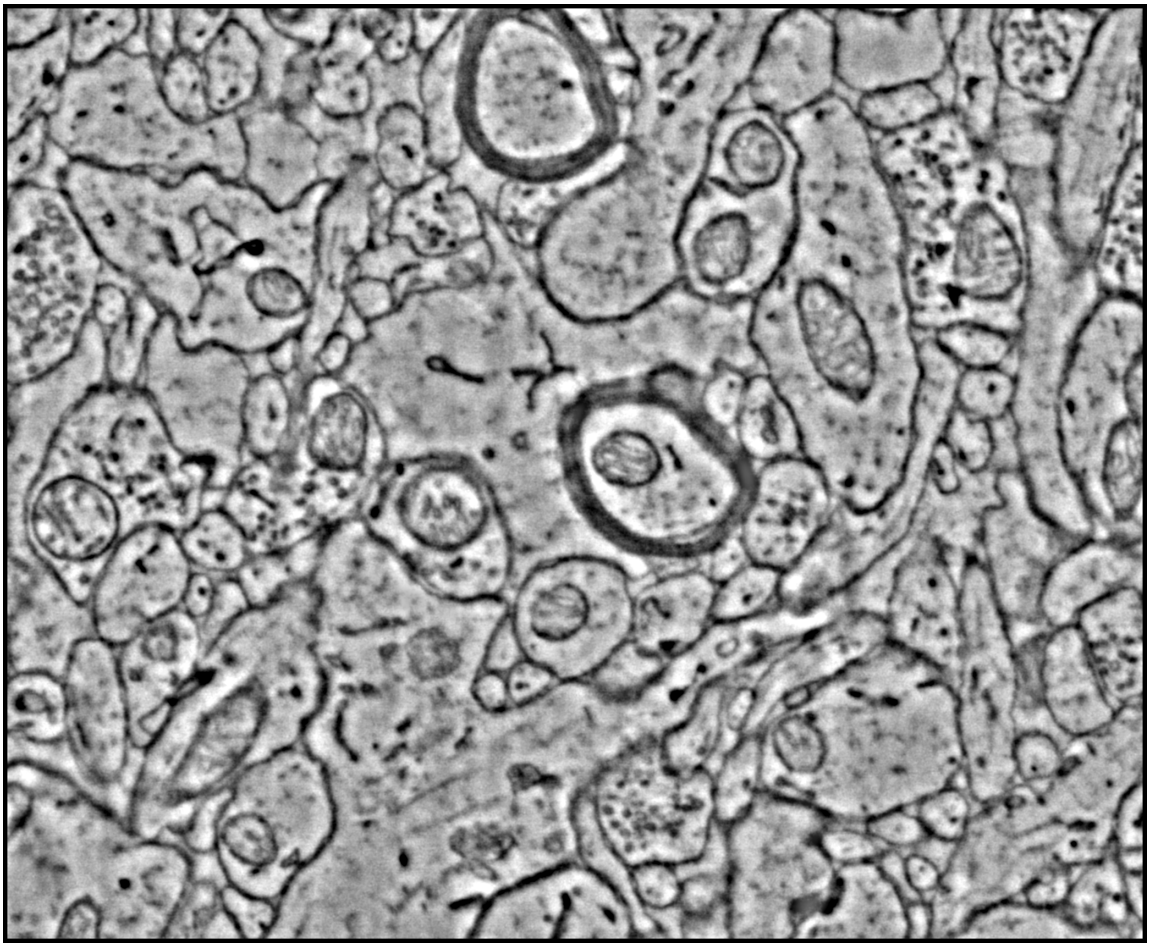
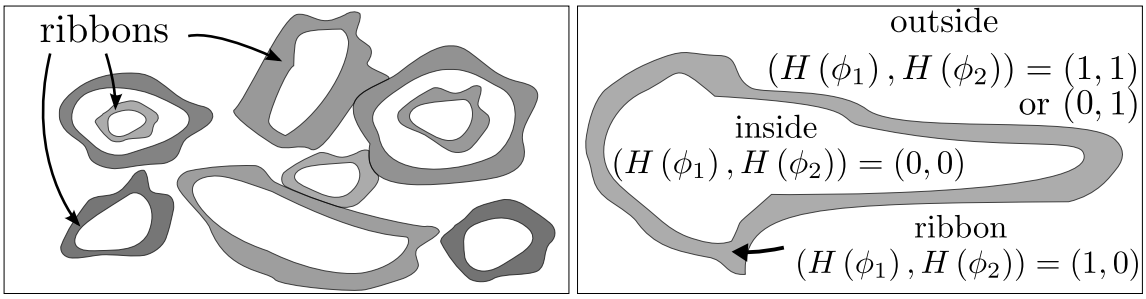


Figure 3.2: *Example of a cross section of a 3D volume of brain tissue acquired with an electron microscope (1px. ≈ 3nm x 3nm). Given the nature of the images, we can make the assumption that sections can be decomposed into a set of "ribbons" of varying thickness.*

terms in the energy. The above energy seeks a partitioning that favors intensity smoothness along each partition and penalizes the size of their boundary sets.

In order to translate the above problem into a multiphase formulation, we use $n = m + 1$ level set functions and follow a similar multi-phase implementation



(a) (b)

Figure 3.3: (a): *We make the assumption that any cross section of the brain tissue can be decomposed into a set of "ribbons".* (b): *In the simplified case where we only have one ribbon, the ribbon partitions the image into three different phases. According to our multiphase encoding of Eq. 3.2, we need at least two distance field functions ϕ_1 and ϕ_2 to represent three phases.*

as [104, 114, 121].

In this framework, a *vector level set function* Φ is defined as $\Phi = (\phi_1, \ldots, \phi_n)$ where each $\phi_i : \Omega \to \mathbb{R}$ is a level set function. Similarly, a partition Ω_i is represented by a binary vector K_i of length n. This way, we can define the characteristic function χ_i for each partition Ω_i as follows:

$$\chi_i = \begin{cases} 1 & \text{when } (H(\phi_1), \ldots, H(\phi_n)) = K_i \\ 0 & \text{otherwise,} \end{cases} \tag{3.2}$$

where $(H(\phi_1), \ldots, H(\phi_n))$ is a binary vector that takes different values when evaluated in different partitions of the image. The function $H(\phi)$ is the Heaviside function introduced in Section 2.3 that takes the values $H(\phi) = 0$ when $\phi \leq 0$ and $H(\phi) = 1$ when $\phi > 0$.

We can now rewrite Eq. 3.1 in terms of the level set functions as:

$$E(\Phi, c) = \sum_{i=1}^{m} \left(\lambda_i \int_\Omega (u - c_i)^2 \chi_i dx + \frac{\alpha_i}{2} \int_\Omega |\nabla \chi_i| \right). \tag{3.3}$$

The Euler-Lagrange equation corresponding to the gradient descent of the functional in Eq. 3.3 yields a system of n evolution equations for $(\phi_1, \ldots, \phi_n)$ (see [60] for an example of a four-phase system).

3.4 Adding Geometric Priors to the Multiphase Framework

The variational formulation of Eq. 3.1 only uses what are conventionally called *internal terms* (those associated with the regularity of the interface), and *data terms* (those associated with the image data). There is nothing in such a formulation that constrains the relative shape and positioning of the partitions. In this section, we present a way of controlling the geometric arrangement of the partitioning by coupling several phases using dynamic force fields. These force fields generate velocities for each partition that we add to those generated by the gradient descent of the Mumford Shah (MS) functional. First, we recall the relationship between curve evolution, level sets and their connection with force fields.

As we saw in Section 2.3, the Level Set Method connects the propagation of a 2D front γ to the evolution of the zero level set of a function $\phi(\gamma, t)$. This way, γ propagates with a speed $\gamma_t \in \mathrm{T}_\gamma M$, if and only if, by the chain rule, ϕ propagates

according to the Level Set equation:

$$\nabla\phi \cdot \gamma_t + \phi_t = 0. \tag{3.4}$$

where $T_\gamma M \in \mathbb{R}^2$ is the tangent space of the manifold M of closed curves immersed in $\mathbb{R}^2$, defined at γ [122]. The n evolution equations for $(\phi_1, \ldots, \phi_n)$ that result from the gradient descent of the MS functional in Eq. 3.3 propagate the zero-level sets of each ϕ function with speeds $\left(\gamma_{1t}^{MS}, \ldots, \gamma_{nt}^{MS}\right)$ due to the connection established by the Level Set Method. Since the encoding of each partition in Eq. 3.2 links the evolution of each level set function with the evolution of each partition, evolution of the level set functions according to gradient descent also implicitly propagates each partition $(\Omega_1, \ldots, \Omega_m)$ with speeds that we denote as $\left(\Gamma_{1t}^{MS}, \ldots, \Gamma_{mt}^{MS}\right)$. These velocities result from the optimization of the minimum partition problem, and therefore have a variational nature.

Consider now a vector field $v : \Omega \to \mathbb{R}^2$. We can build a velocity field γ_{it}^v for the zero-level set of one of our functions ϕ_i if we project v on $T_{\gamma_i} M$ via the mapping $\gamma_{it}^v = v \cdot \gamma_i^\perp$. Such mapping extracts the normal component of v to the zero level set of ϕ_i. As in the Mumford Shah case, the vector field v implicitly also maps into a vector of velocities $\left(\Gamma_{1t}^v, \ldots, \Gamma_{mt}^v\right)$ for each of the boundary sets of the partitions.

We can extend this concept to build a *force field* $F = (v_1, \ldots, v_n)$ for each zero-level set γ_i. In the more general case, we can generate a number of force fields that, if designed wisely, could be used to arrange the different partitions $(\Omega_1, \ldots, \Omega_m)$ on the plane. Since $T_\gamma M$ is a vector space, the velocity fields

derived from such force fields can be added to the velocities derived from the MS functional as follows:

$$\gamma_t = \gamma_t^{MS} + \sum_{j=1}^{p} \mu_j \gamma_t^{F_j}, \tag{3.5}$$

where p is the number of force fields used, $\gamma_t = (\gamma_{1t}, \ldots, \gamma_{nt})$ is the vector of total velocities of the zero-level sets, $\gamma_t^{MS} = \left(\gamma_{1t}^{MS}, \ldots, \gamma_{nt}^{MS}\right)$ is the vector of velocities given by the gradient descent of the MS functional, and $\gamma_t^{F_j} = \left(\gamma_{1t}^{v_{1j}}, \ldots, \gamma_{nt}^{v_{nj}}\right)$ is the vector of velocities given by the action of the force field $F_j = \left(v_{1j}, \ldots, v_{nj}\right)$ onto each of the ϕ functions. The parameters μ_j determine the strength of the force fields relative to that arising from the MS functional.

It is important to note, however, that since the vector fields v do not have to be irrotational (curl-free), some of them might not equal the gradient of a scalar potential, and therefore it is not always possible to guarantee the existence of an equivalent variational formulation for each vector field. For this reason, we will consider that a solution for the segmentation problem is found (the evolution finishes) when the velocities from the optimization of the MS functional and those from the external force fields balance each other and an equilibrium is reached ($||(\gamma_{1t}, \ldots, \gamma_{nt})||_{L^2} \leq \epsilon$). In the best case, the velocities from the MS functional balance their counterparts from the force fields F_j:

$$\gamma_t^{MS} = -\sum_{j=1}^{p} \mu_j \gamma_t^{F_j}. \tag{3.6}$$

In the next section we show several examples of force fields that can be used to induce geometrical arrangements in the partitioning.

3.5 Active Ribbons

Consider the problem of segmenting neural processes on a stack of electron micrographs. Neural membranes may appear thicker or thinner for different neural processes in these images (e.g., see Fig. 3.2 for an example of a section in a stack). Some axons may be myelinated (covered by an electrically insulating layer) which gives them the look of having a much thick membrane around them than the rest. Moreover, the analysis of the thickness of membranes can reveal important information about the connectivity of biological neural networks and neural-related diseases [123]. Rather than just segmenting the cell with a single one-pixel wide boundary as with standard level-set methods, it may be useful to segment and extract cellular membranes in their own partition in the segmentation. Moreover, adding prior knowledge about the arrangement of the desired partitions may make the segmentation problem better-posed. These observations motivate our interest in modeling membrane-looking or "ribbon-looking" partitions.

We start by defining an Active Ribbon as the deformable region between two non-intersecting contours, one contained within the other. Figure 3.3(a) shows that a single active ribbon yields a partitioning of the image in three regions (*inside*, *ribbon* and *outside*). According to our multiphase encoding we discuss in Section 3.3, we need at least two distance field functions ϕ_1 and ϕ_2 to represent three regions.

We now consider three force fields that can be used to arrange the image partitions into a set of ribbons. The first two forces control the shape of each

ribbon and their ability to find the right cellular boundaries in an image of brain tissue. The third one is required for problems where we wish to track simultaneously multiple neurons through several sections of an EM stack, and models the interaction between ribbons by controlling their mutual repulsion or attraction.

Force Field for Ribbon Consistency

Consider the force field:

$$F_1 := (v_1, v_2) = \left(\frac{\nabla \phi_2}{\sigma_1}, -\frac{\nabla \phi_1}{\sigma_2} \right) \tag{3.7}$$

where v_1 and v_2 act on ϕ_1 and ϕ_2, respectively, and σ_1 and σ_2 are scalar fields defined on the image plane. The joint action of v_1 and v_2 creates a repulsion or an attraction force between the boundaries of the ribbon depending on the sign of σ_1 and σ_2 (see Fig. 3.4).

Ideally, we want to design the ribbon so that it shows plasticity (no resistance to deformation) when the thickness of the ribbon is within some reasonable range (5-20 nm). In such cases, the evolution of the ribbons would be mostly driven by the gradient flow of the MS functional of Eq. 3.1. On the other hand, we want to trigger the repulsion or attraction between the boundaries of the ribbon when the ribbon has an abnormally small or large thickness respectively.

Following this reasoning, we can design σ_2 and σ_1 for ϕ_1 and ϕ_2 so that they react to the proximity between the boundaries of the ribbons. We can achieve this effect by setting $\sigma_1(\phi_2)$ and $\sigma_2(\phi_1)$ with a profile such as the one depicted in Fig.

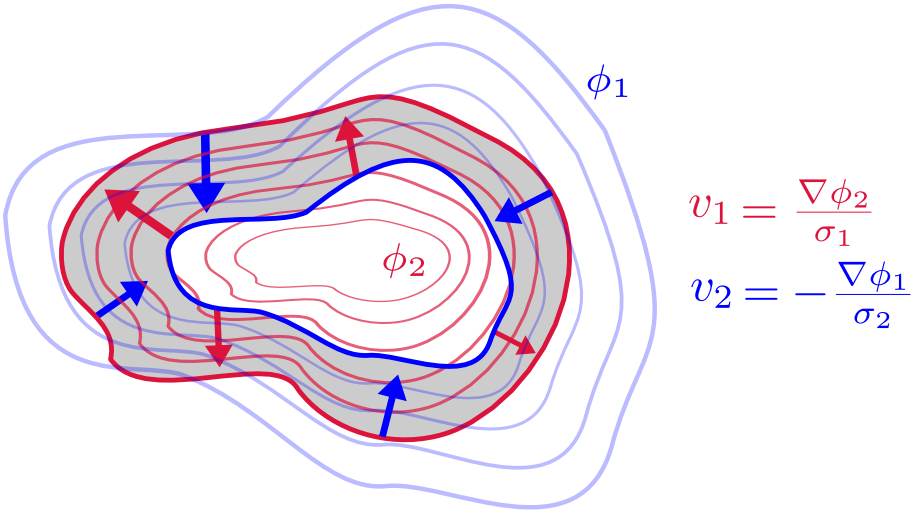


Figure 3.4: *Force field for ribbon consistency. The zero iso-contours of ϕ_1 and ϕ_2 (the inner and outer outlines of the ribbon respectively) are depicted in dark blue and dark red respectively. The non-zero iso-contours of ϕ_1 and ϕ_2 are also depicted in light blue and light red lines, respectively. We compute the gradient of ϕ_1 on the outer boundary of the ribbon (i.e., the zero iso-contour of ϕ_2), and the gradient of ϕ_2 on the inner boundary of the ribbon (i.e., the zero iso-contour of ϕ_1), to define $v_1 = \frac{\nabla \phi_2}{\sigma_1}$ and $v_2 = -\frac{\nabla \phi_1}{\sigma_2}$ respectively (both $\nabla \phi_1$ and $\nabla \phi_2$ point away from the ribbon, as both ϕ_1 and ϕ_2 are negative inside and positive outside each contour). This way v_1 pushes ϕ_1 towards ϕ_2, and v_2 pushes ϕ_2 towards ϕ_1. Depending on the sign of σ_1 and σ_2, the force field $F_1 = (v_1, v_2)$ would attract or repel the outer and inner boundaries of the ribbon.*

3.5. A piecewise polynomial approximation of this profile is discussed in Section 3.6.

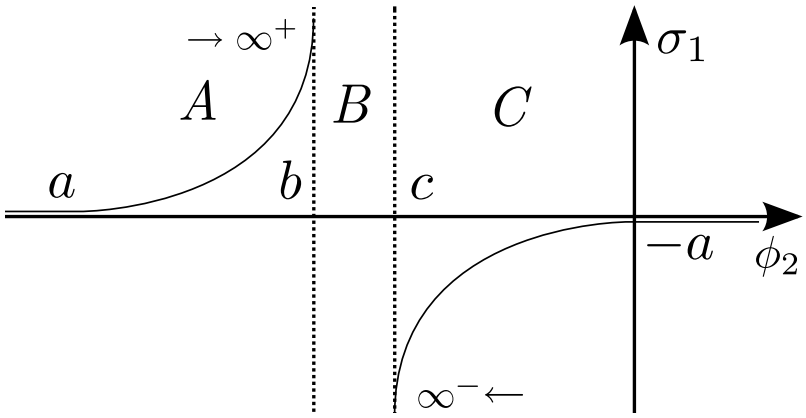


Figure 3.5: *σ_1 as a function of ϕ_2 The graph for σ_2 is obtained by reversing the ϕ axis ($\sigma_2(\phi) = \sigma_1(-\phi)$). The region labeled as A corresponds to the attraction ($\sigma_1 \geq 0 \to v_1 \geq 0$) between the ribbon boundaries (stronger as the boundaries separate from each other). Region B is the region where the ribbon shows a plastic behavior (no resistance towards deformation). Region C corresponds to the repulsion ($\sigma_1 \leq 0 \to v_1 \leq 0$) between the ribbon boundaries (stronger as the boundaries come closer to each other).*

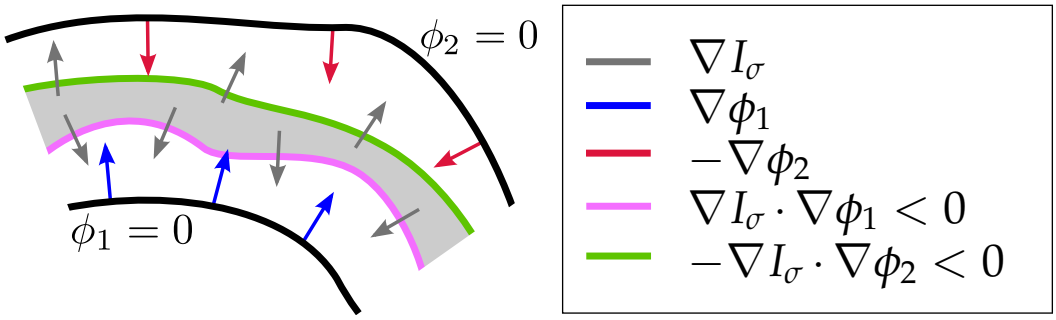
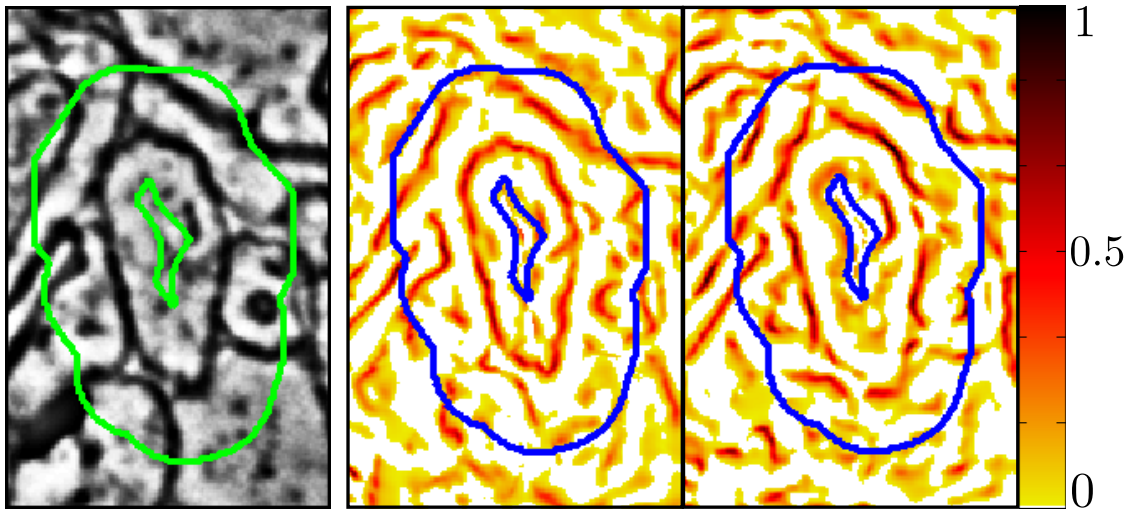


Figure 3.6: *We want the outer side of the active ribbon ($\phi_2 = 0$) to be attracted towards the outer side of the cellular membrane (feature map/bitmap in green), and the inner side of the ribbon ($\phi_1 = 0$) towards the inner cellular membrane (feature map/bitmap in pink).*



(a) (b)

Figure 3.7: (a)*: Active ribbon (in green) on a sample image.* (b)*: Normalized feature maps (bitmaps)* e_1 *(left) and* e_2 *(right) for the inner and outer side of the ribbon (blue).*

Force Field for Ribbon-cell Interaction

The previous force field model adds a purely geometric force to the multiphase partitioning that induces the creation of elastic ribbons. The combination of such force field with the MS functional can segment the image in homogeneous ribbon-like partitions. However, the model so far does not necessarily guarantee that the two different boundaries of the ribbon will be attracted to different boundaries of the cellular membranes (see Fig. 3.6). In this section we introduce a second force field that, when added to the previous model, achieves precisely that.

We start by recalling the vector field convolution model introduced in [66]. Given a feature map defined on the image $e : \Omega \to \mathbb{R}^+$ (i.e., a bitmap of the same

size of the image where higher values indicate the presence of some image feature, such as edges) we can build a smooth vector field $v : \Omega \to \left(v_x, v_y\right) \in \mathbb{R}^2$ that points towards the highest values in e as:

$$\left(v_x, v_y\right) = e * k = \left(e * k_x, e * k_y\right), \tag{3.8}$$

where $k := \left(k_x, k_y\right)$ is a 2D user-defined vector field kernel such as the one shown in Fig. 3.8(b), and the subscripts refer to the x and y components of each vector field. See [66] for additional notes on kernel selection.

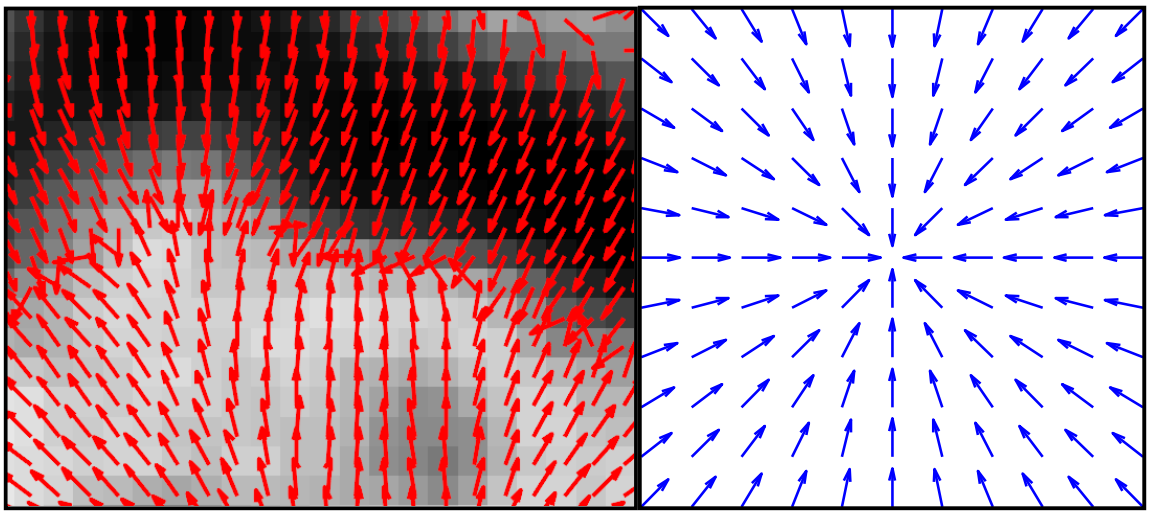
Building on this idea, consider the following force field:

$$\begin{aligned} F_2 := (v_1, v_2) &= (e_1 * k, e_2 * k) \\ &= \left(\left(\nabla I_\sigma \cdot \nabla \phi_1\right) * k, -\left(\nabla I_\sigma \cdot \nabla \phi_2\right) * k\right), \end{aligned} \tag{3.9}$$

where v_1 and v_2 act on ϕ_1 and ϕ_2, respectively, I_σ is a smoothed version of the image I, and $e_1 = \nabla I_\sigma \cdot \nabla \phi_1$ and $e_2 = -\nabla I_\sigma \cdot \nabla \phi_2$ are the feature maps for ϕ_1 and ϕ_2, respectively. The above force field is made of a vector field v_1 that acts on ϕ_1 by pushing its zero-level set towards the inner side of the cellular boundaries, and v_2 that acts on ϕ_2 by pushing its zero-level set towards their outer side (see Fig. 3.6). Figures 3.7 and 3.8(a) show the feature maps and the resulting force field on examples of real images.

Interaction Between Ribbons

The previous two force fields control the shape and segmentation of each ribbon in an individual manner, and therefore cannot accommodate interaction between



(a) (b)

Figure 3.8: (a): *Close-up of the vector field v_1 of the force field F_2 generated from the ribbon and sample image in Fig. 3.7 (a). The vector field v_1 points towards the inner side of the cellular membranes (v_2, not shown in this figure, points towards the outer side).* (b): *The kernel we used for computing the vector fields that push each side of ribbon towards the right side of the cell membranes. The vector fields are obtained by computing the convolution of this kernel we the feature maps.*

neighboring ribbons. However, going back to our application of interest (segmenting and tracking neural boundaries across sections), it is known that when moving from one section to the next one through the 3D volume of brain tissue, the cellular boundaries of the neurons should not change their relative position abruptly. That is, if two neurons were adjacent to each other and/or relatively close in one section, they should be so in the next one.

In this section we show how, using force fields, we can also control such relative geometric arrangement ribbons. This may be used to avoid undesired

segmentations and speed up the convergence of the multiphase evolution.

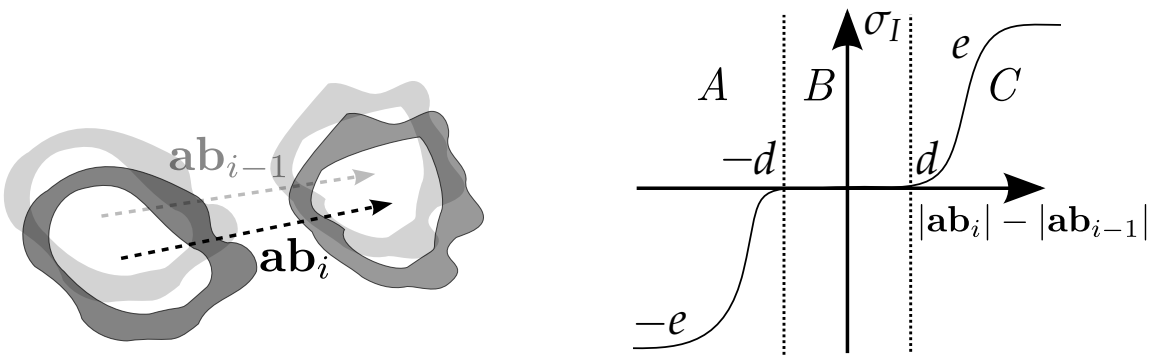
Consider that while processing section i of the volume of brain tissue we are given the segmentation results from section $i-1$. Since we know the location of each ribbon in section $i-1$, we can build a force field $F_{ab} := (v_1, v_2, v_3, v_4)$ for each pair of ribbons a and b, such that:

$$\begin{aligned} v &= \sigma_I \left(|\mathbf{ab}_i| - |\mathbf{ab}_{i-1}| \right) \hat{\mathbf{ab}}_i \\ v_1 &= v_2 = v, \quad v_3 = v_4 = -v \end{aligned} \quad , \tag{3.10}$$

where the vector fields v_1 and v_2 act on the level set functions ϕ_1 and ϕ_2 defining ribbon a, while v_3 and v_4 act on the functions ϕ_3 and ϕ_4 defining the ribbon b. The vectors $\mathbf{ab}_i$ and $\mathbf{ab}_{i-1}$ point from the center of mass of the ribbon a to the one of ribbon b on the sections $i-1$ and i, and $\sigma_I : \mathbb{R} \to \mathbb{R}$ is a function that controls the strength of the mutual repulsion (when $\sigma_I < 0$) or attraction (when $\sigma_I > 0$) between the two ribbons (see Fig. 3.9). Section 3.6 discusses a piecewise polynomial approximation of σ_I.

3.6 Experiments

In this section we present several experiments that show the robustness of Active Ribbons on real data. First, we compare our model with three other well known level set-based deformable models for the segmentation of single cellular boundaries. We then compare our method with a geometrically-unconstrained multiphase level set model for the segmentation multiple cellular boundaries. Finally, we show how our active ribbon model can efficiently track cellular



(a) (b)

Figure 3.9: (a)*: A force of mutual repulsion or attraction is created between each pair of ribbons based on the result from previous sections.* (b)*: σ_I is chosen to create a elastic repulsion (region A) or attraction force (region C) to keep the distance between the ribbons within some range (region B).*

boundaries on a sequence of cross-sections obtained from a volume of brain tissue and give an example of failed segmentation.

Figure 3.10 shows a comparison of our model with several geometrically-unconstrained deformable models. In the best cases, these models were able to accurately extract the outer cellular boundaries or the inner cellular boundaries alone, but none of them could segment the cellular membranes in a single isolated partition, making them invalid for example, for the analysis and study of membrane thickness. Figure 3.11 shows a gradual comparison between a classical geometrically-unconstrained multiphase level set model and our active ribbon model.

We note that, by using two level set functions to represent each ribbon in our model, we allow ribbons to overlap on the image. This overlap is intentional and

it is a consequence of our partition encoding, where the zero-level sets of multiple level set functions can intersect. When modeling Active Ribbons, our encoding uses a total of $2m$ level set functions for m ribbons, but gives direct access to the contour of each ribbon via their zero-level sets. Such encoding also guarantees that ribbons can share cellular boundaries and agrees with the visual appearance of neural membranes when they are imaged with electron microscopy (see Fig. 3.2 for examples). Finally, such overlap facilitates the tracking of neurons throughout the volume of brain tissue, since this way each ribbon can more easily sit on image boundaries.

Figure 3.12 shows the tracking of a neural process, where in each section, the active ribbon model is initialized with the results obtained in the previous cross-section of the brain tissue. Such approach to tracking only works a neural process is relatively orthogonal to the image plane, since otherwise neural processes would experiment large displacements between consecutive sections.

Finally, Fig. 3.13 shows an example of a failed segmentation with one ribbon, where the presence of multiple adjacent cellular membranes confused the ribbon to believe it was segmenting a single process.

Setting the parameters of the model

We show a summary of the parameters of our model in Table 3.1. We adjusted the model experimentally using the following set up for the results shown in Figs. 3.10, 3.11, 3.12 and 3.13. For the ribbon-consistency force field we used $|b| = 10$ pixels and $|c| = 25$ pixels and $a = 10^{-9}$ to adjust the thickness of the

ribbon with σ_1 and σ_2. For the Mumford-Shah weighting parameters we used $\alpha_i = \lambda_i = 1$ for the foreground partitions and 0 for the background partitions. For the parameters controlling the strength of the force fields on the ribbons we used $\mu_1 = 1.5$, $\mu_2 = \mu_3 = 1$ giving a slightly stronger weight to the ribbon-consistency field than to the other two forces. Finally, for the ribbon-to-ribbon interaction, we adjusted the plasticity σ_I using $d = 50$ pixels and $e = 2$ pixels.

In practice we noticed the model to be fairly robust to changes in μ, α and λ, the parameters controlling the balance between the Mumford Shah active contour model and the force fields (up to $\sim$50% of variation). The model was a bit more sensitive to the adjustment of σ_1 and σ_2 which control the thickness of the ribbon and the strength of the coupling of the outer with the inner contours. More specifically, we noticed that if this coupling was too weak, the ribbon could develop protrusions and sometimes "steal pixels" from neighboring cell membranes. However, the sensitivity of our model to all these parameters was small in comparison to the dependency to the initial conditions (the starting location and shape of the ribbon), a problem shared by many deformable models based on level sets [57]. If the ribbon is not properly initialized, it can fail to segment a cell correctly, as we show in Fig. 3.13.

Finally, we note that some of the parameters of the model should be adjusted to the specifics of the data. For example, the desired ribbon width in pixels should depend on the chosen imaging resolution (the thickness of the cell membranes in pixels) and perhaps the preparation of the tissue.

Parameter	Description
$\sigma_1, \sigma_2, \sigma_I$	Desired ribbon width and elasticity between ribbons
α, λ	Mumford-shah parameters (ribbon smoothness vs. image fitting)
μ_1,μ_2,μ_3	Relative strength between the Mumford-Shah flow and our force fields

Table 3.1: *Summary of the parameters of our model and their description.*

3.7 Summary

The work presented in this chapter is, to the best of our knowledge, the first effort to show a direct way of geometrically constraining a multiphase level sets flow for image segmentation. This is done using dynamic force fields built with vector fields such as those introduced previously in the literature for active contours for helping them deal with local minima. Our method requires no training set and can be easily combined with other variational level set segmentation models.

When used interactively, Active Ribbons can ask the user to draw a ribbon on the screen for a given neural process on a section and then attempt to reconstruct the rest of the process automatically through the stack (Figs. 3.12 and 3.13 show examples of Active Ribbons operating in this way). This idea served as the foundation foundation of NeuroTrace, a computer program developed in our lab published in [124] and [45] and that extended Active Ribbons to work on GPU hardware and on 2D planes with arbitrary 3D orientation. This facilitates the tracing of neural processes that are not necessarily orthogonal to the image plane.

While interactive tools such as NeuroTrace can help neuroscientists reconstruct neurons in 3D faster than by using manual tools such as Reconstruct, requiring the user to provide an initial segmenting ribbon for each process can be time

consuming when working with large stacks. Inspired by the need to automate segmentation further, in the next chapter we explore a method for segmenting arbitrarily-long image sequences automatically.

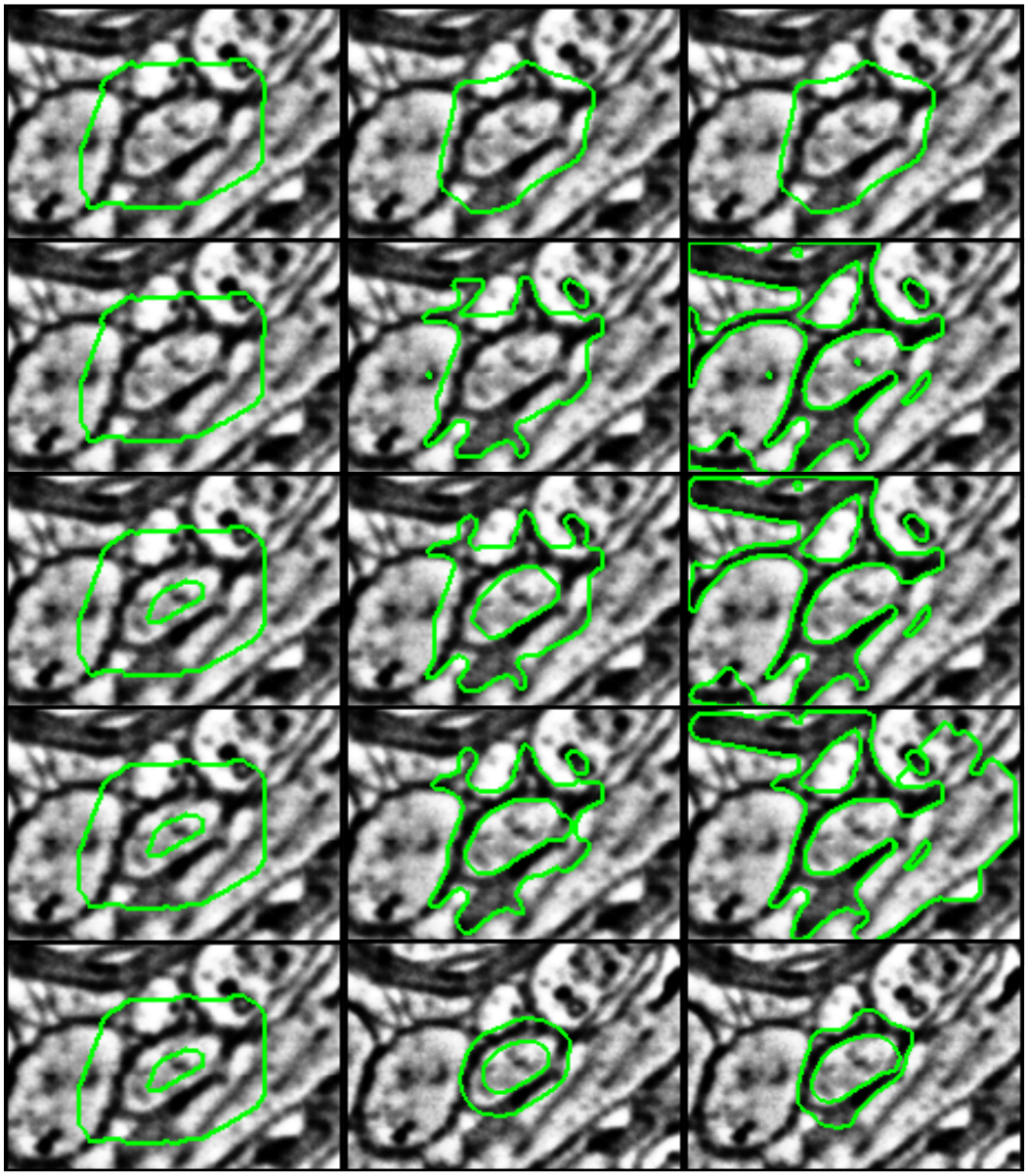


Figure 3.10: ***First row:*** *Best result obtained using the models of [66] and [67] with the deformable model initialized with a single connected component from outside.* ***Second and third row:*** *Results obtained using the two-phase model of [60] when the deformable model was initialized from outside, and from both inside and outside, respectively; Similar results were obtained with this initialization for the models of [66] and [67].* ***Fourth row:*** *Results obtained with the model of [116] with region descriptors based on the mean and variance for both the foreground and background.* ***Fifth row:*** *Results obtained with our active ribbon model. The columns correspond to iterations 1, 34 and 71.*

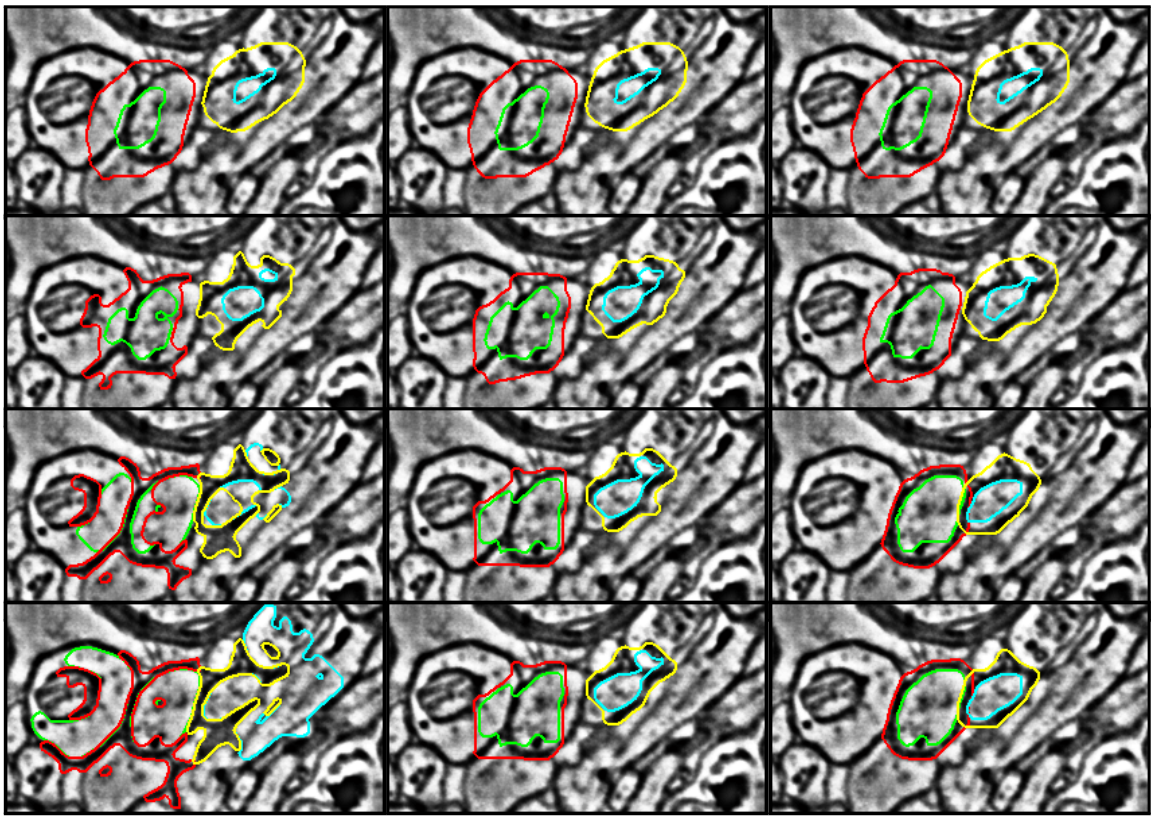


Figure 3.11: *Gradual column-by-column comparison between a classical geometrically-unconstrained multiphase level set model and our active ribbon model.* ***First column:*** *Results obtained with the model of [60] with λ_i set to 0 for the background phases in Eq. 3.3.* ***Second column:*** *Results obtained with our force field for ribbon consistency.* ***Third column:*** *Results obtained with all the force fields discussed enabled. The columns correspond to iterations 1, 34, 55 and 79.*

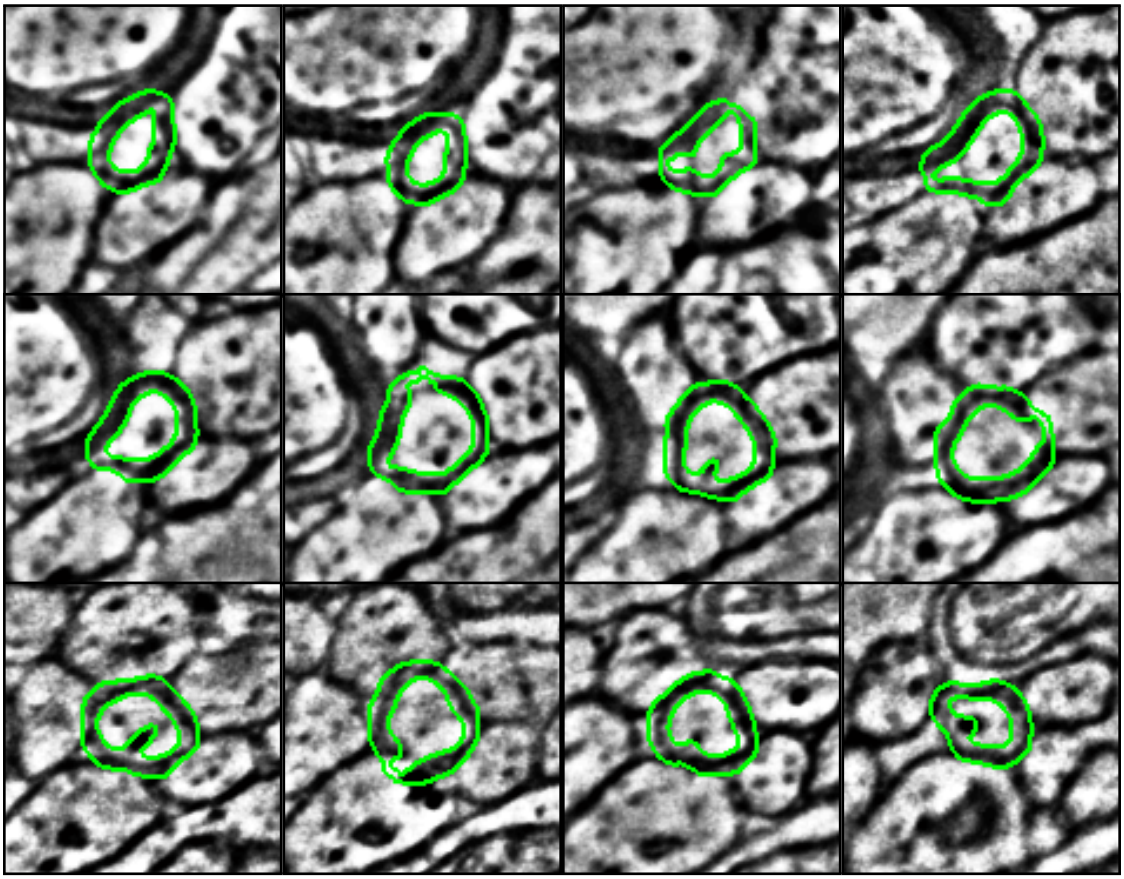


Figure 3.12: ***Left-to-right and top-to-bottom****: Tracking of a cellular membrane through 12 sections. The active ribbon is initialized with the results obtained in the previous cross-section of the brain tissue.*

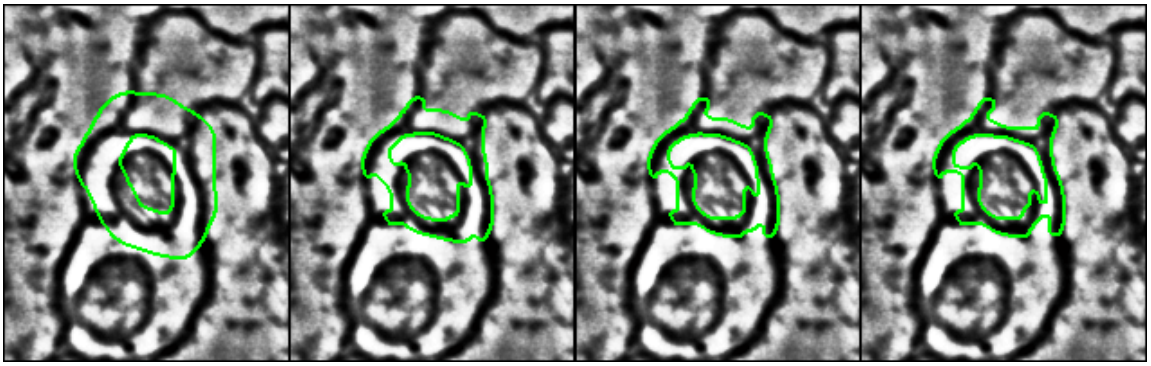


Figure 3.13: *Example of a failed segmentation with one ribbon. The images correspond to iterations 1, 10, 16 and 30.*

CHAPTER 4

Automatic Segmentation of Image Sequences with MHVS

4.1 Synopsis

In the previous chapter we showed an extension of level set methods, a well-discussed technique in the image segmentation literature, for the problem of segmenting cell membranes of neural processes. We introduced a framework that allows for the control of the geometric arrangement of image segments, and showed how to use it to design Active Ribbons, a deformable geometric model for segmenting neural processes in EM stacks. By providing an initial estimate of a segmenting ribbon on an initial section, Active Ribbons can iteratively segment a process in each section and iteratively trace it through the stack.

However, throughout the previous chapter we assumed that an initial location of the ribbon is provided on a given section. This can be done, for example, by an user that draws the initial shape of the ribbon on the screen. Also, as we discussed in Section 2.3, the energy functional in Eq. 3.1 is non-convex [60] and

level set methods are can get trapped in local minima during the gradient descent optimization [61].

In this chapter we take a different path. Motivated by the connection between connectomics and video segmentation described in Section 1.3, we consider a problem closely related to reconstructing neurons in arbitrary large EM stacks, the problem of unsupervised *on-line* video segmentation.

We focus on the on-line photometric segmentation of videos, namely the automatic labeling of video based on texture, color and/or motion features. By on-line segmentation we refer to the problem of segmenting a video in a continuous, sequential manner, i.e., loading only a few frames at a time. This is different from *real-time* segmentation, which is the segmentation under hard time constraints.

Instead of using level set methods for segmenting video, we explore the applicability of the statistical graphical models discussed in Section 2.4, and propose Multiple Hypothesis Video Segmentation (MHVS), a video segmentation method based on the notion of *deferred inference* [125], where segmentation decisions are deferred until more frames are received.

As we show in this chapter, MHVS segments arbitrarily long video streams by considering a few frames at a time, and it handles the automatic creation, continuation and termination of labels with no user initialization or supervision.

MHVS works by first generating several pre-segmentations per frame and enumerating multiple possible trajectories of pixel regions within a sliding time window. After assigning each trajectory a score, MHVS lets the trajectories

Figure 4.1: *Results from the on-line, unsupervised, photometric segmentation of a video sequence with MHVS.* ***Top****: original frames.* ***Bottom****: segmented frames. MHVS keeps track of multiple possible segmentations, collecting evidence across several frames before assigning a label to every pixel in the sequence. It also automatically creates and terminates labels depending on the scene complexity and as the video is processed.*

compete with each other to segment each window. We determine the solution to this segmentation problem as the MAP labeling of a higher-order Markov Random Field (MRF) known as the *Robust Potts model*, which we briefly mentioned in Section 2.4. This framework allows MHVS to achieve spatial and temporal long-range label consistency while operating in an on-line manner.

We test MHVS on several videos of natural scenes with arbitrary camera and object motion. The results included in the chapter were published in [126].

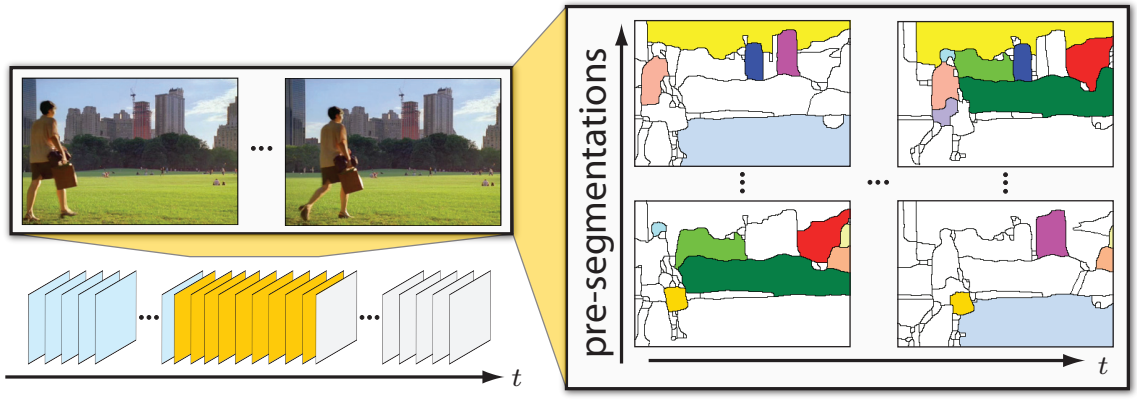


Figure 4.2: ***Left****: MHVS labels a video stream in an on-line manner considering several frames at a time.* ***Right****: For each processing window, MHVS generates multiple pre-segmentations per frame, and finds sequences of superpixels (shown as colored regions) that match consistently in time. Each of these sequences, called a superpixel flow, is ranked depending on its photometric consistency and considered as a possible label for segmentation. The processing windows overlap one or more frames to allow labels to propagate from one temporal window to the next.*

4.2 Background

Unsupervised photometric video segmentation, namely the automatic labeling of a video based on texture, color and/or motion, is an important computer vision problem with applications in areas such as activity recognition, video analytics, summarization, surveillance and browsing [127, 128]. However, despite its significance, the problem remains largely open for several reasons.

First, the unsupervised segmentation of arbitrarily long videos requires the automatic creation, continuation and termination of labels to handle the free flow

of objects entering and leaving the scene. Due to occlusions, objects often merge and split in multiple 2D regions throughout a video. Such events are common when dealing with natural videos with arbitrary camera and object motion. A complete solution to the problem of multiple-object video segmentation requires tracking object fragments and handling splitting or merging events.

Second, robust unsupervised video segmentation must take into account spatial and temporal long-range relationships between pixels that can be several frames apart. Segmentation methods that track objects by propagating solutions frame-to-frame [129, 130] are prone to overlook pixel relationships that span several frames.

Finally, without knowledge about the number of objects to extract from an image sequence, the problem of unsupervised video segmentation becomes strongly ill-posed [131]. Determining the optimal number of clusters is a fundamental problem in unsupervised data clustering [131].

Contributions

MHVS is, to the best of our knowledge, the first solution to the problem of unsupervised on-line video segmentation that can effectively handle arbitrarily long sequences, create and terminate labels as the video is processed, and still preserve the photometric consistency of the segmentation across several frames.

Although the connections between tracking and video segmentation are well discussed in e.g., [129, 130, 132–134], we present the first extension of the idea of deferred inference from Multiple Hypothesis Tracking (MHT) [135, 136] to the

problem of unsupervised, multi-label, on-line video segmentation.

MHVS relies on the use of space-time segmentation hypotheses, corresponding to alternative ways of grouping pixels in the video. This allows MHVS to postpone segmentation decisions until evidence has been collected across several frames, and to therefore operate in an on-line manner while still considering pixel relationships that span multiple frames. This extension offers other important advantages. Most notably, MHVS can dynamically handle the automatic creation, continuation and termination of labels depending on the scene complexity, and as the video is processed.

We also show how higher-order Markov Random Fields, which we use to solve the hypothesis competition problem, can be applied to the problem of unsupervised on-line video segmentation. Here, we address two important challenges. First, the fact that only a subset of the data is available at any time during the processing, and second, that the labels themselves must be inferred from the data. A working example of MHVS is illustrated on Fig. 4.1.

Related Work

We briefly discuss previous work on video segmentation and show why most existing methods are not directly applicable to the problem of unsupervised photometric on-line video segmentation.

A large number of video segmentation methods assume that the number of labels is known *a priori* or is constant across frames [137–142]. While this assumption is appropriate for certain segmentation problems such as

foreground-background video segmentation [140–142], the numbers of objects in a video is generally unknown and hard to estimate. Only a few methods can adaptively and dynamically determine the number of labels required to segment a video photometrically [130, 133]. Such ability to adjust is especially important in on-line video segmentation, since the composition of the scene tends to change over time.

Another common assumption is that all frames are available at processing time and can be segmented together [132, 141, 143–146]. While this assumption holds for certain applications, the segmentation of arbitrarily long video sequences (e.g., such as large EM stacks in connectomics) requires the ability to segment and track results by working with a few frames at a time, for example, by processing them in an on-line fashion. Unfortunately, those methods that can segment video in an on-line manner usually track labels from frame to frame [129, 130, 133] (i.e.,, they only consider two frames at a time), which makes them sensitive to segmentation errors that gradually accumulate over time. Moreover, a large number of these methods, have only been applied to the problem of background-foreground binary segmentation [129] or rely on object detection for segmenting the video.

A large part of the literature is dedicated to the problem of interactive video segmentation. As we discussed in Section 1.4 these methods require the user to provide graphical input in the form of scribbles, seeds, or even accurate boundary descriptions in one or multiple frames to initiate or facilitate the segmentation [142, 143]. This can be helpful or even necessary to obtain the

desired grouping of segments or pixels, but in this chapter we aim for an automatic segmentation method.

4.3 An Overview of MHVS

The three main steps in MHVS are: hypotheses enumeration, hypotheses scoring, and hypotheses competition.

A *hypothesis* refers to one possible way of grouping several pixels in a video, i.e.,, a correspondence of pixels across multiple frames. More specifically, we define a hypothesis as a grouping or flow of *superpixels*, where a superpixel refers to a contiguous region of pixels obtained from a tessellation of the image plane without overlaps or gaps. This way, each hypothesis can be viewed as a possible label that can be assigned to a group of pixels in a video (see Fig. 4.2)

Since different hypotheses represent alternative trajectories of superpixels, hypotheses will be said to be *incompatible* when they overlap; that is, when one or more pixels are contained in more than one hypothesis. In order to obtain a consistent labeling of the sequence, we aim for the exclusive selection of only one hypothesis for every set of overlapping hypotheses (see an example in Fig. 4.3).

Depending on the photometric consistency of each hypothesis, we assign them a score (a likelihood). This allows us to rank hypotheses and compare them in probabilistic terms. The problem of enumeration and scoring of hypotheses is discussed in Section 4.4. Once hypotheses have been enumerated and assigned a score, we make them compete with each other to label the video sequence. This

competition penalizes the non-exclusive selection between hypotheses that are incompatible in the labeling. In order to resolve the hypotheses competition problem, MHVS relies on MAP estimation on a higher-order Markov Random Field. In this probabilistic formulation, hypotheses will be considered as labels or classes that can be assigned to superpixels on a video. Details about this step are covered in Section 4.5.

For the segmentation of arbitrarily long video sequences, the above process of hypotheses enumeration, scoring and competition is repeated every few frames using a sliding window. By enumerating hypotheses that include the labels from the segmentation of preceeding windows, solutions can be propagated sequentially throughout an arbitrarily long video stream.

4.4 Enumeration and Scoring of Hypotheses

The enumeration of hypotheses is a crucial step in MHVS. Since the number of all possible space-time hypotheses grows factorially with frame resolution and video length, this enumeration must be selective. The pruning or selective sampling of hypotheses is a common step in the MHT literature, and it is usually solved via a "gating" procedure [147].

We address the enumeration and scoring of hypotheses in two steps. First, we generate multiple pre-segmentations for each frame within the processing window using segmentation methods from the literature, e.g., [18, 148]. Then, we match the resulting segments across the sequence based on their photometric

Figure 4.3: *Two hypotheses that are incompatible. The hypotheses (shown in green and red) overlap on the first two frames. The segmentation of the sequence should encourage their exclusive selection. MHVS ranks hypotheses photometrically and penalizes the non-consistent selection of the most coherent ones over time.*

similarity. Those segments that match consistently within the sequence will be considered as hypotheses (possible labels) for segmentation.

The above approach can be modeled with a Markov chain of length equal to that of the processing window. This allows us to look at hypotheses as time sequences of superpixels that are generated by the chain, with the score of each hypothesis given by the probability of having the sequence generated by the chain.

We formalize this approach as follows. Given a window of F consecutive frames from a video stream, we build a weighted, directed acyclic graph $G = (V, E)$ that we denote as a *superpixel adjacency graph*. In this graph, a node

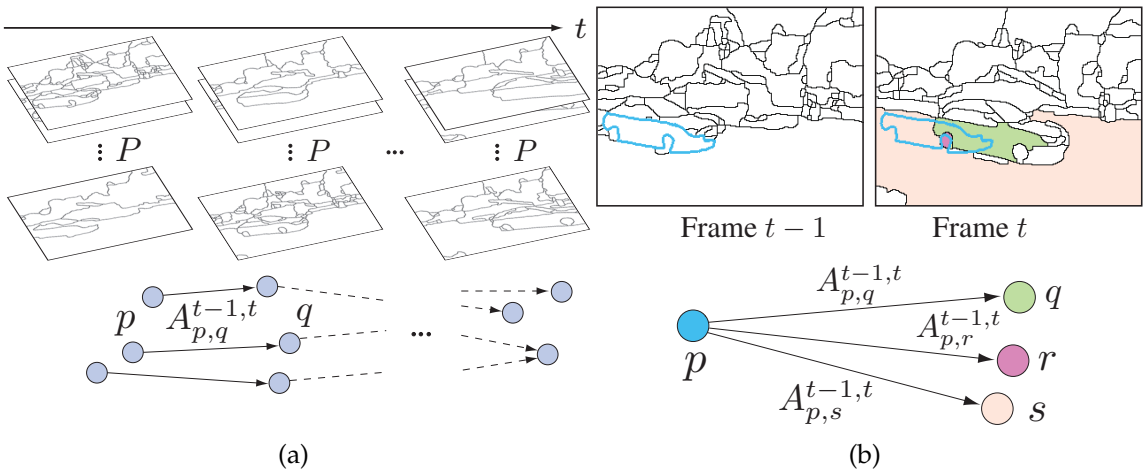


Figure 4.4: *Construction of the superpixel adjacency graph for the enumeration of hypotheses (flows of superpixels).* (a) *For each processing window, MHVS generates P pre-segmentations on each frame. Each of them groups pixels at different scales and according to different photometric criteria. The nodes in the graph represent superpixels from some of the pre-segmentations on each frame, and the edges capture the photometric similarity between two temporally adjacency superpixels.* (b) *Two superpixels are considered to be temporally adjacent if they overlap spatially but belong to two different and consecutive frames.*

represents a superpixel from one of the pre-segmentations on some frame within the processing window, and an edge captures the similarity between two temporally adjacent superpixels (superpixels that overlap spatially but belong to two different and consecutive frames). Edges are defined to point from a superpixel from one of the pre-segmentations on time t to a superpixel from one of the pre-segmentations on $t + 1$. Fig. 4.4 shows an illustration of how this graph is built.

The above graph can be thought as the transition diagram of a Markov chain of length F [81]. In this model, each frame is associated with a variable that represents the selection of one superpixel in the frame, and the transition probabilities between two variables are given by the photometric similarity between two temporally adjacent superpixels. By sampling from the chain, for example, via ancestral sampling [81] or by computing shortest paths in the transition diagram, we can generate hypotheses with strong spatio-temporal coherency.

More specifically, for a given window of F frames, and the set of all superpixels $\mathcal{V} = \{V_1, \ldots, V_F\}$ generated from P pre-segmentations on each frame, we can estimate the joint distribution of a sequence of superpixels $(z_1, \ldots, z_F)$ as

$$p(z_1, \ldots, z_F) = p(z_1) \cdot \prod_{t=2}^{t=F} A_{j,k}^{t-1,t}, \tag{4.1}$$

where the transition matrices $A_{j,k}^{t-1,t}$ capture the photometric similarity between two temporally adjacent superpixels $z_{t-1} = j$ and $z_t = k$, and are computed from the color difference between two superpixels in LUV colorspace, as suggested in [149]. In order to generate hypotheses that can equally start from any superpixel on the first frame, we model the marginal distribution of the node z_1 as a uniform distribution. Further details about the generation of pre-segmentations and the sampling from the Markov chain are discussed in Section 4.6.

Once a set of hypotheses has been enumerated, we measure their temporal coherency using the joint distribution of the Markov chain. Given a set of L

hypotheses $\mathcal{H} = \{H_1, \ldots, H_L\}$, we define the score function $s : \mathcal{H} \rightarrow [0, 1]$ as:

$$s\left(H_k\right) = N_1 \cdot p\left(z_1 = v_1, \ldots, z_F = v_F\right) = \prod_{t=2}^{F} A^{t-1,t}_{v_{t-1},v_t}, \tag{4.2}$$

where $(v_1, \ldots, v_F)$ is a sequence of superpixels comprising a hypothesis H_k and N_1 is the number of superpixels on the first frame.

Propagation of solutions The above approach needs to be extended to also enumerate hypotheses that propagate the segmentation results from preceding processing windows. We address this problem by allowing our processing windows to overlap one or more frames. The overlap can be used to consider the superpixels resulting from the segmentation of each window when enumerating hypothesis in the next window. That is, the set of pre-segmented superpixels $\mathcal{V} = \{V_1, \ldots, V_F\}$ in a window w, $w > 1$, is extended to include the superpixels that result from the segmentation of the window $w - 1$.

4.5 Hypotheses Competition

Once hypotheses have been enumerated and scored for a particular window of frames, we make them compete with each other to label the sequence. We determine the solution to this segmentation problem as the MAP labeling of a random field defined on a sequence of fine grids of superpixels. This framework allows us to look at hypotheses as labels that can be assigned to random variables, each one representing a different superpixel in the sequence (see Fig. 4.5).

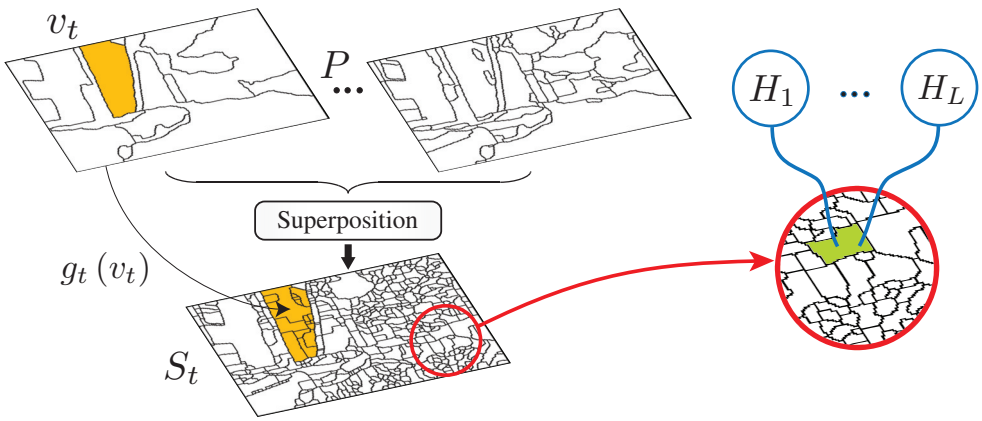


Figure 4.5: *We define our higher-order conditional random field on a sequence of fine grids of superpixels $\mathcal{S} = \{S_1, \ldots S_F\}$. Each grid S_t is obtained as the superposition of the P tessellations that were generated for the enumeration of hypotheses. The mapping g_t takes superpixels v_t from one of the pre-segmentations to the superposition S_t. Each superpixel in S_t is represented in our MRF with a random variable that can be labeled with one of the hypotheses $\{H_1, \ldots, H_L\}$.*

Our Markov Random Field consists of three potentials. A unary potential that measures how much a superpixel within the MRF grid agrees with a given hypothesis, a binary potential that encourages photometrically similar and spatially neighboring superpixels to select the same hypothesis, and a higher-order potential that forces the consistent labeling of the sequence with the most photometrically coherent hypotheses over time (See Fig. 4.6 for an illustration).We formalize this as follows. For each processing window of F frames, we define a random field of N variables x_i defined on a sequence of grids of superpixels $\mathcal{S} = \{S_1, \ldots S_F\}$, one for each frame. Each grid S_t is obtained as the

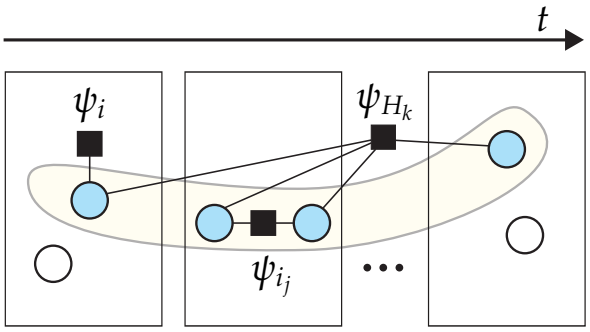


Figure 4.6: *An illustrative factor graph of the higher-order random field that we define for segmenting a sliding time window. Each rectangle represents a different frame. The yellow shape represents a hypothesis (a possible label) H_k for segmentation. The hypothesis H_k touches several superpixels of the MRF grid (represented as blue circles). Each such superpixel is associated with a random variable that may choose H_k as their label for the segmentation. The unary, pairwise and higher-order potentials, ψ_i, $\psi_{i,j}$ and ψ_{H_k}, respectively, encourage neighboring superpixels that are photometrically similar to choose high-scoring hypotheses.*

superposition of the P pre-segmentations used for the enumeration of hypotheses, and yields a mapping g_t that takes every superpixel from the pre-segmentations to the set S_t (see Fig. 4.5). The random variables x_i are associated with superpixels from $\mathcal{S}$, and take values from the label set $\mathcal{H} = \{H_1, \ldots, H_L\}$, where each hypothesis H_k is sampled from the Markov chain described in the previous section.

A *labeling* of the random field, is an assignment $\mathbf{x} \in \mathcal{H}^N$ of labels (hypotheses) to the N random variables. We recall from Eqs. 2.11 and 2.10 in Section 2.4, that

the MAP labeling $\mathbf{x}^*$ our Markov Random Field takes the form:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{H}^N} p\left(\mathbf{x}\right) = \arg\max_{\mathbf{x}\in\mathcal{H}^N} \frac{1}{Z}\prod_{c\in\mathcal{C}} \psi_c\left(\mathbf{x}_c\right), \tag{4.3}$$

where $\psi(\mathbf{x}_c)$ are potentials of exponential form $\psi(\mathbf{x}_c) = \exp\left\{\sum_{\alpha\in\alpha_c} \theta_\alpha \phi_\alpha(\mathbf{x}_c)\right\}$ defined on cliques of variables c from some set $\mathcal{C}$, and θ_c are weighting parameters between the different potentials. The labeling $\mathbf{x}_c$ represents the assignment of the random variables x_i within the clique c to their corresponding values in $\mathbf{x}$.

We next define three different types of potentials ψ_c for our objective function in Eq. 4.3. Following the standard notation in the literature [77,93,150], we define the potentials as penalties on the labeling, i.e., functions that assign lower probabilities to the posterior distribution to undesirable labelings of the MRF. The idea is to obtain the solution that minimizes such penalties by computing the maximum *a posteriori* (MAP) solution.

We model the potentials to encourage the consistent photometric labeling of the sequence. The unary potentials favor the selection of hypotheses that provide a high detail (fine) labeling of each frame. The pairwise potentials encourage nearby superpixels to get the same label if they are photometrically similar. Finally, the higher-order potentials encourage the exclusive selection of hypotheses that are incompatible with each other.

Unary potentials. We use the mappings $g = (g_1, \ldots, g_F)$ between the pre-segmentations and the grids S_t (see Fig. 4.5) to define the penalty of assigning a hypothesis x_i to the random variable X_i representing the superpixel s_i. To do this, we define the unary potential (i.e., a potential acting on single variables)

$\psi_i(x_i) = \exp\{-\theta_u \phi_i(x_i)\}$, where $\phi_i(x_i)$ is a penalty function for each variable, such that:

$$\phi_i(x_i) = 1 - d(s_i, g(x_i)), \tag{4.4}$$

where $\phi_i(x_i)$ penalizes the variable X_i if it chooses a hypothesis x_i that does not overlap well with the associated superpixel s_i. The function $g(x_i)$ represents the mapping of the superpixels that define the hypothesis x_i to the set of superpixels $\mathcal{S}$, and $\theta_u > 0$ defines the weight of the unary potentials in Eq. 4.3. The function $d(a, b)$ measures the Dice coefficient $\in [0, 1]$ on the plane between the sets of pixels a and b (the spatial overlap between a and b, normalized to a value between 0 and 1), and is defined as $d(a, b) = 2|a \cap b| / (|a| + |b|)$.

Since the set of superpixels $\{S_1, \ldots S_F\}$ represents an over-segmentation on each frame (it is obtained from a superposition of tessellations), the unary potential favors labelings of the sequence with spatially thin hypotheses, i.e., those with the highest overlap with superpixels on the MRF grid, in the Dice-metric sense. To see this, note that the higher the overlap between a hypothesis and an MRF superpixel, the higher the dice-coefficient d in Eq. 4.4, and the higher the probability in Eq. 4.3.

Pairwise potentials. We define a pairwise potential $\psi_{i,j}(x_i, x_j) = \exp\{-\theta_p \phi_{i,j}(x_i, x_j)\}$ for every pair of spatially adjacent superpixels s_i, s_j in each frame, such that:

$$\phi_{i,j}(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j \\ 1 - b(i, j) & \text{otherwise,} \end{cases} \tag{4.5}$$

where $\theta_p > 0$ defines the weight of the pairwise potentials in Eq. 4.3, and $b(i,j) \in [0,1]$ captures the photometric difference between adjacent superpixels. In practice, we obtain the value of $b(i,j)$ by reading from a boundary map of the image (see Section 4.6 for details).

Note the similarity between the potential of Eq. 4.5 and the pairwise potentials of the Ising and Potts models that we discussed in Section 2.4. The potential of Eq. 4.5 is a type of piecewise-constant model that encourages a discontinuity-preserving labeling of the video by penalizing label disagreement between neighboring superpixels that are photometrically similar [50]. To see this, note that, the more photometrically different two superpixels are, the lower the penalty in the pairwise potential, and the higher the probability in Eq. 4.3, when they choose different labels (hypotheses).

Higher-order Potentials. As mentioned in Section 4.3, we penalize the non-exclusive selection of hypotheses that are incompatible with each other. To do this, we design a higher-order potential that favors the consistent selection of the most photometrically coherent hypotheses over time. The notion of label consistency was formalized by Kohli et al. in [93] and [150] with the introduction of the Robust P^n model (also known as the Robust Potts model), which they applied to the problem of supervised multi-class image segmentation. Here, we use this model to penalize label disagreement between superpixels comprising hypotheses of high photometric coherency. For each hypothesis H_k, we define the potential $\psi_{H_k}(\mathbf{x}_k) = \exp\{-\theta_h \phi_{H_k}(\mathbf{x}_k)\}$, where $\phi_{H_k}(\mathbf{x}_k)$ is a penalty function

such that:

$$\phi_{H_k}(\mathbf{x}_k) = \begin{cases} N_k(\mathbf{x}_k)\,\frac{1}{Q_k}s(H_k) & \text{if } N_k(\mathbf{x}_k) \leq Q_k \\ s(H_k) & \text{otherwise,} \end{cases} \tag{4.6}$$

where $\mathbf{x}_k$ represents the labeling of the superpixels comprising the hypothesis H_k, $N_k(\mathbf{x}_k)$ denotes the number of variables not taking the dominant label (i.e.,, it measures the label disagreement within the hypothesis), and $\theta_h > 0$ defines the weight of the higher-order potentials in Eq. 4.3. The score function $s(H_k)$ defined in the previous section measures the photometric coherency of the hypothesis H_k (see Eq. 4.2). The truncation parameter Q_k controls the rigidity of the higher-order potential [93], and we define it as:

$$Q_k = \frac{1 - s(H_k)}{\max\limits_{m \in [1,L]} (1 - s(H_m))} \cdot \frac{|c|}{2}. \tag{4.7}$$

The function $\phi_{H_k}(\mathbf{x}_k)$ assigns higher penalties to labelings where the superpixels within a high-scoring hypothesis choose different labels. In other words, the more photometrically coherent a hypothesis is, the higher its score, and the higher the penalty for disagreement between the labels of the MRF superpixels overlapping with it. See Fig.4.8(a) for an illustration of the potential.

Labeling. Once we have defined unary, binary and higher-order potentials for our objective function in Eq. 4.3, we approximate the MAP estimate of the MRF using a graph cuts solver for the Robust P^n model [93]. This solver relies on a sequence of α-expansion moves that are binary, quadratic and submodular, and therefore exactly computable in polynomial time [93] (see Section 2.4 for more details on how to solve such MAP-MRFs problems). From the association

between variables x_i and the superpixels in S, the MAP estimate also yields the segmentation of all the pixels within the processing window.

Handling mergers and splits. The implicit (non-parametric) object boundary representation provided by the random field [50] allows MHVS to easily handle merging and splitting of labels over time; when an object is split, the MAP labeling of the graph yields disconnected regions that share the same label (see Fig. 4.7). Since labels are propagated across processing windows, when the parts come back in contact, the labeling yields a single connected region with the same label. The automatic merging of object parts that were not previously split in the video is also implicitly handled by MHVS. This merging occurs when the parts of an object are included within the same hypothesis (i.e. one of the pre-segmentations groups the parts together).

In order to create new labels for parts of old labels, when the parts become distinguishable enough over time to be tracked, a final mapping of labels is done before moving to the next processing window. We handle this scenario by comparing the spatial overlap between new labels (from the current processing window) and old labels (from the preceding processing window). We check for new labels l that significantly overlap spatially with some old label p, but barely overlap with any other old label q. We can measure such overlaps using their Dice coefficients, and we denote them by γ_p and γ_q. Then, if $\gamma_p > \gamma_1$ and $\gamma_q < \gamma_2, \forall q \neq p$, for a pair of fixed parameters $\gamma_1, \gamma_2 \in [0, 1]$, we map the label l to p, otherwise l is considered a new label (see Fig. 4.8(b) for an example).

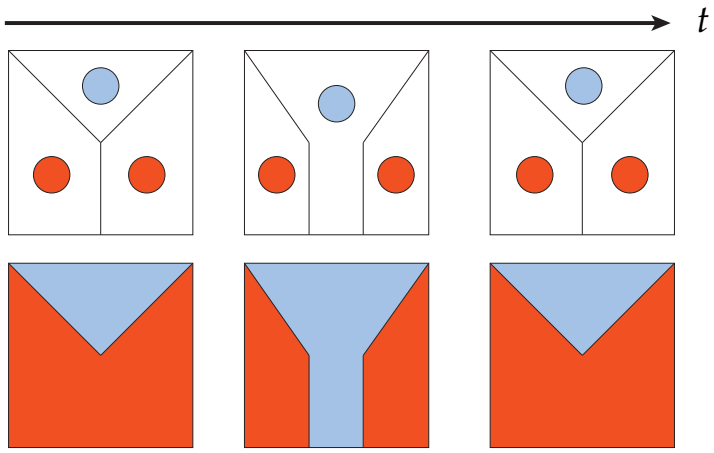


Figure 4.7: *Illustration of the segmentation of three consecutive frames. The use of an MRF on a grid of superpixels (top) that encodes the labeling of pixels in each frame enables MHVS to handle objects that may split and merge over time (e.g., the object in red).*

4.6 Experimental Results

Most previous work on unsupervised photometric video segmentation has focused on the segmentation of sequences with relatively static backgrounds and scene complexity [130, 132, 138, 141]. In this work, however, we show results on natural videos with arbitrary motion on general scenes. Since existing datasets of manually-labeled video sequences are relatively short (often less than 30 frames), and usually contain a few number of labeled objects (often only foreground and background), we collected five videos of outdoor scenes with 100 frames each, and manually annotated an average of 25 objects per video every three frames. The videos include occlusions, objects that often enter and leave the scene, and dynamic backgrounds (see Figs. 4.1 and 4.9 for frame examples).

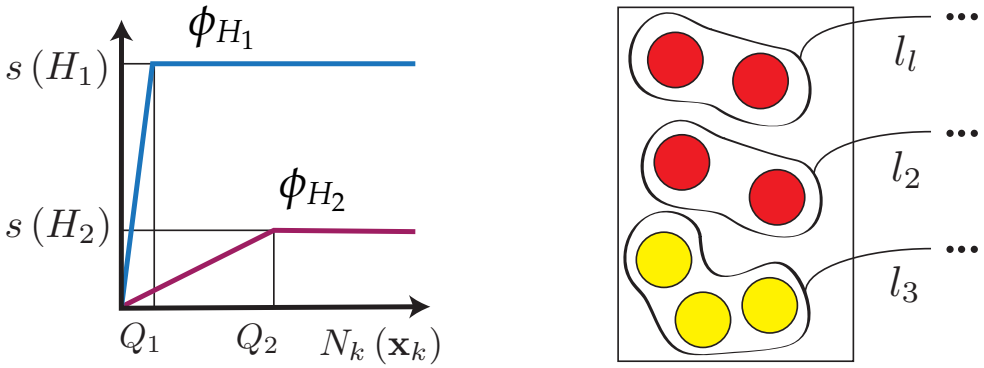


Figure 4.8: (a) *Higher-order penalty (y-axis) as a function of label disagreement within a hypothesis (x-axis) for two overlapping hypotheses H_1 and H_2, with H_1 being more photometrically coherent than H_2. The function ϕ_{H_1} strongly penalizes any label disagreement within H_1, while ϕ_{H_2} tolerates significantly higher label disagreement within H_2.* (b) *The colored circles represent superpixels that were labeled in the preceding processing window (each color being a different label). The groupings l_1, l_2 and l_3 are the result of the MAP labeling within the current processing window. Depending on the selection of γ_1 and γ_2 (see text), l_1 and l_2 are considered as new labels or mapped to the label depicted in red.*

We compared MHVS with spatio-temporal mean-shift (an *off-line* method, similar to [144]), and pairwise graph propagation (an on-line method with frame-to-frame propagation, similar to [130]). In both methods we included color, texture and motion features. For the test with mean-shift, each video was processed in a single memory-intensive batch. For our MHVS tests, F was set to 5 frames to meet memory constraints, but values between 3 and 10 gave good results in general. The size of the processing window was also observed to

balance MHVS's ability to deal with strong motion while preserving long-term label consistency. We used an overlap of one frame between processing windows and generated $P = 30$ pre-segmentations per frame using the gPb boundary detector introduced by Maire et al. [148], combined with the OWT-UCM algorithm from [151] to obtain a hierarchy of 2D pre-segmentations.

As mentioned in Section 4.4, hypotheses can be obtained via ancestral sampling [81] (i.e., sampling from the conditional multinomial distributions in the topological order of the chain), or by computing shortest paths in the transition diagram from each superpixel on the first frame to the last frame in the window (i.e., computing the most likely sequences that start with each value of the first variable in the chain). We follow this second approach. Neither guarantees that every MRF superpixel is visited by a hypothesis. In our implementation, such MRF superpixels opt for a dummy (void) label, and those that overlap with the next processing window are later considered as sources for hypotheses. The parameters α_e weighting the relative importance between the unary, pairwise and higher-order potentials in Eq. 4.3 were set to 10, 2 and 55, respectively, although similar results were obtained within a 25% deviation from these values. The pairwise difference between superpixels $b(i, j)$ was sampled from the boundary map generated by OWT-UCM and the parameters γ_1 and γ_2 that control the mapping of new labels to old labels were set to 0.8 and 0.2, respectively.

We measured the quality of the segmentations using the notion of *segmentation covering* introduced by Arbeláez et al. in [151]. The covering of a human

Method	Video 1	Video 2	Video 3	Video 4	Video 5
MHVS (multi-frame on-line)	**0.62**	**0.59**	**0.45**	**0.54**	0.42
Graph propagation (pairwise on-line)	0.49	0.37	0.36	0.39	0.34
Mean-shift (off-line)	0.56	0.39	0.34	0.38	**0.44**

Table 4.1: *Best segmentation covering obtained with MHVS, pairwise graph propagation and mean-shift across five outdoor sequences that were manually annotated. Frame examples from Video 1 are shown in Fig. 4.1, and from Videos 2 to 5 in Fig. 4.9, top to bottom. Higher segmentation coverings are better.*

segmentation S by a machine segmentation S', can be defined as:

$$C\left(S' \rightarrow S\right) = \frac{1}{N} \sum_{V \in S} |V| \cdot \max_{V' \in S'} d\left(V, V'\right), \tag{4.8}$$

where N denotes the total number of pixels in the video, and $d\left(V, V'\right)$ is the Dice coefficient in 3D between the labeled spatio-temporal volumes V and V' within S and S', respectively. These volumes can possibly be made of multiple disconnected space-time regions of pixels. Table 4.1 shows the values of the best segmentation covering achieved by each method on our five videos.

Running time. The unary, pairwise and higher-order potentials of Eq. 4.3 are sparse. Each random variable (representing an over-segmented superpixel) overlaps few other hypotheses. No overlap makes the unary and higher-order terms associated with the hypothesis zero. The pre-segmentations, enumeration of hypotheses and measuring of photometric similarities between superpixels can be parallelized, and each processing window must be segmented (Eq. 4.3 solved) before moving to the next processing window. With this, in our tests, MHVS run on the order of secs/frame using a Matlab-CPU implementation.

Figure 4.9: ***Top to fourth row****: Results from the on-line, unsupervised, photometric segmentation of four video sequences of varying degrees of complexity with MHVS. The examples show MHVS's ability to adjust to changes in the scene, creating and terminating labels as objects enter and leave the field of view.* ***Fourth and fifth row:*** *Comparison between MHVS (fourth row) and pairwise graph propagation (based on [130]) (fifth row). The frames displayed are separated by 5-10 frames within the original segmented sequences.*

4.7 Summary and Discussion

MHVS is, to the best of our knowledge, the first solution to the problem of fully unsupervised on-line video segmentation that can segment videos of arbitrary length, with unknown number of objects, and effectively manage object splits and mergers. Our framework is general and can be combined with any image segmentation method for the generation of space-time hypotheses. Alternative scoring functions, to the ones presented here, can also be used for measuring photometric coherency or similarity between superpixels.

We believe this work bridges further the gap between video segmentation and tracking. It also opens the possibility of integrating the problem of on-line video segmentation with problems in other application domains such as event recognition or on-line video editing. Future work could include extensions of MHVS based on on-line learning for dealing with full occlusions and improving overall label consistency.

In our tests with MHVS, we observed that labels sometimes have a short lifespan in the segmentation output, resulting in a perceptual flickering of labels in the segmented video. We attribute this to mostly two reasons. First, the fact that it is difficult to find matching superpixels in 2D pre-segmentations of consecutive frames, and second, that the Robust Potts model encourages but does not guarantee exclusivity between hypotheses.

Regarding the first observation, we have noticed that gPb-OWT-UCM [151], the algorithm that we use to obtain 2D pre-segmentations on each frame (currently one of the state-of-the-art algorithms in the segmentation of natural

images), can produce significantly different 2D segmentations on consecutive frames even when only a few pixels change between them. We attribute this in part to the fact that OWT-UCM (the algorithm that builds the segmentations from the pixel output of gPb) is designed to build a stack of 2D segmentations on each frame with the requirement that the 2D segments on the frame can be nested to produce a hierarchy of segments. We have observed that such hierarchies can change abruptly between consecutive frames, sometimes moving the same segment from the top to the bottom level of the hierarchy, or viceversa. This is the case even though the boundary pixel detector (gPb) on which gPb-OWT-UCM is based, seems to provide a robust response across frames.

Regarding the problem of exclusivity between hypotheses, the Robust Potts model allows for parts of hypotheses to win during the competition between hypotheses, i.e., it encourages the exclusive selection of hypotheses using "soft constraints". However, the higher-order potential encourages but does not *guarantee* exclusivity. This has some advantages, such as allowing for the optimization to produce combinations of partial hypotheses when labeling the video, i.e., as opposed to constraining the optimization to make hard choices between hypotheses that may overlap only for a few pixels. However, the model also allows for overlapping hypotheses to steal pixels from each other if they both have similarly high scores, which may result in label flickering.

In the next chapter, we build on the ideas behind MHVS and propose Segmentation Fusion (or just Fusion), a statistical graphical model for the segmentation of connectomic stacks. As in MHVS, Fusion combines multiple

possible 2D pre-segmentations for obtaining a 3D segmentation (a 3D connectomic stack representing a video in MHVS). However, in contrast to MHVS, Fusion does not rely on the explicit discovery or enumeration of possible labels for segmentation, and uses a variational formulation with hard integer programming constraints that forces exclusivity between segments on each section.

CHAPTER 5

Segmentation Fusion for Connectomics

5.1 Synopsis

In this chapter we propose a novel method for the automatic 3D neuron reconstruction that, like MHVS, also combines multiple 2D segmentations per section of brain tissue to obtain the final 3D segmentation. Unlike previous efforts in connectomics, where the reconstruction is usually done on a section-to-section basis, or by the agglomerative clustering of 2D segments, we leverage information from the entire volume to obtain a globally optimal 3D segmentation.

Additionally, and in contrast to MHVS (Chapter 4) the method in this chapter does not require an exhaustive search of possible segment trajectories to obtain a segmentation of the stack that is globally optimal in the space of all possible groupings of available segments in the stack.

To do this, we formulate segmentation as the solution to a fusion problem. We

first enumerate multiple possible 2D segmentations for each section in the stack, and a set of 3D links that may connect segments across consecutive sections. We then identify the fusion of segments and links that provide the most globally consistent segmentation of the stack. We show that this two-step approach of pre-enumeration and posterior fusion yields significant advantages and provides state-of-the-art reconstruction results.

Finally, we also introduce a robust rotationally-invariant set of features, the Adapted Zernike features, that we use to learn and enumerate the above 2D segmentations. Our features outperform previous connectomic-specific descriptors without relying on a large set of heuristics or manually designed filter banks.

Both methods Fusion and the Adapted Zernike features were published in [14].

5.2 Background

In the previous chapter we addressed the problem of on-line segmentation of natural videos, and proposed MHVS, a method that leverages multiple 2D pre-segmentations on each frame to identify labels that can be used to segment the video using a sequential sliding-window. As we recall from the previous chapter, the segmentation pipeline in each window of frames can be summarized in three steps:

- The computation of multiple 2D pre-segmentations for each frame using gPb-OWT-UCM [151], a combination of state-of-the-art boundary detector

with a method that produces a hierarchical segmentation of each frame.

- The enumeration of hypotheses, long-term trajectories of 2D segments, that can be used as labels in the video.
- The computation of the maximum *a posteriori* labeling of a random field known as the Robust Potts model that encourages pixels within each trajectory of segments to get the label associated with the trajectory.

We next recall some properties and observations about MHVS that motivate the work we present in this chapter.

From MHVS to Fusion

In MHVS the set of labels that are available for segmentation is given by the set of trajectories of 2D segments that is pre-enumerated between frames. In Section 4.4, we discussed two possible enumeration schemes, ancestral sampling and shortest paths on the superpixel adjacency graph. However, both enumeration schemes require an exhaustive search in practice if one aims to have MHVS find the best labeling among all possible segment groupings between frames. As we see in this chapter, with Fusion we take a different approach, and model labels as the result of the connected component labeling of a graph that we obtain as a result of an binary integer programming problem. This allows Fusion to indirectly search in the space of all possible labels that can be obtained by grouping segments.

In MHVS we used the Robust Potts model of [93] to encourage the consistent selection of hypotheses with "soft constraints". While, as we noted in Section 4.7, these soft constraints provide some flexibility in the segmentation, they also come at a cost, by allowing high-scoring hypothesis (labels) to "steal" pixels from each other if they overlap. In Fusion, we define a new graphical model fundamentally different from the Robust potts model, where all variables are binary, and there are no soft constraints.

In our experiments with MHVS in the last chapter, we obtained the 2D pre-segmentations for each frame using gPb-OWT-UCM [151]. However, as other authors have also noted [27], gPb was conceived to rely heavily on changes in color and texture to detect image boundaries and does not work well on ssEM images. As we recall from earlier chapters, in ssSEM different cells can be distinguished by the dark membranes (dark thin structures) that separate them, and not by perceptual properties of the intra-cellular pixels that are different from cell to cell. In this chapter we propose our own pixel classifier to obtain 2D pre-segmentations for connectomic datasets. Our pixel classifier aims directly for cell membrane detection and is based on novel set of rotationally-invariant features called the Adapted Zernike features.

Finally, and as we recall from Section 4.7, OWT-UCM requires its output to be a hierarchical 2D segmentation, which we noticed was not stable over time in our experiments. In this chapter, we instead use a sequence of watershed transformations [152] at different heights to obtain the 2D pre-segmentations per section and the 2D segments do not have be able to fit in a hierarchy.

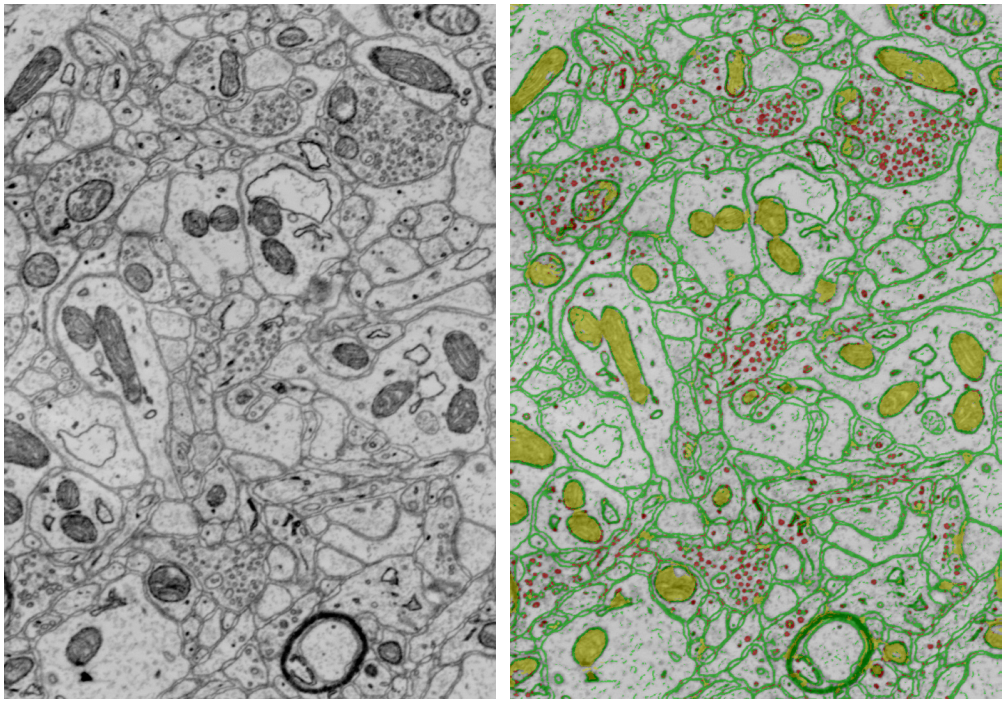


Figure 5.1: *Automatic labeling (right) of cellular structures of a 2D section of a volume of brain tissue (left) using our features. Each pixel is labeled by a random forest classifier trained with the features we introduce in Section 5.5. Mitochondria are shown in yellow, vesicles in red and cellular boundaries in green. The rest of pixels are labeled as cellular background (shown with a transparent label).*

Contributions

We introduce the notion of segmentation fusion, the global fusion of 2D segments and 3D links for the problem of 3D neuron segmentation. Our method compares multiple possible 2D segmentations and linking choices across sections to find the best fusion of segments and links that together form each neuron. Our

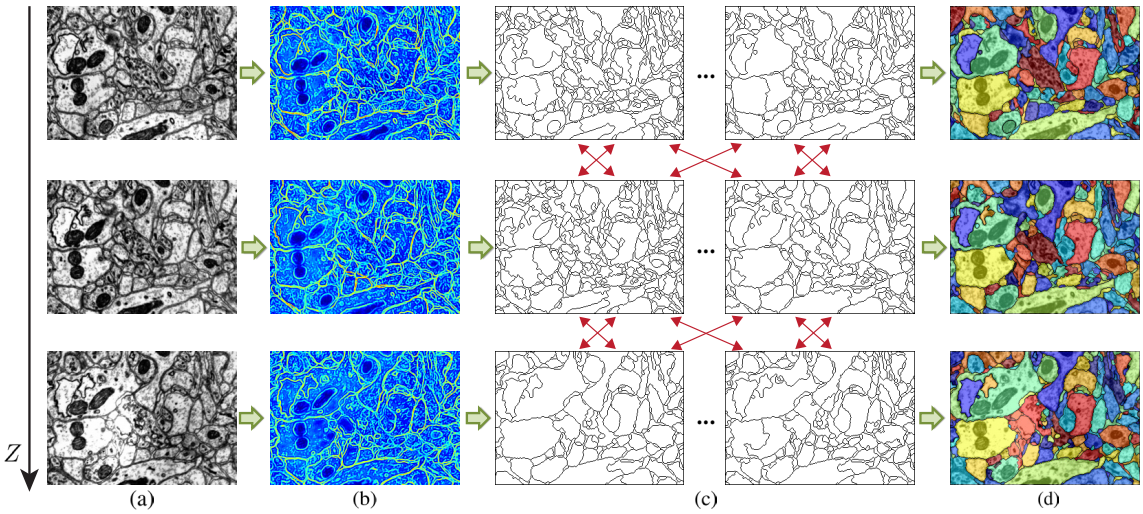


Figure 5.2: *Each section from an input stack of electron microscopy images is represented by each row in the figure, from the original image (column **(a)**), we compute the probability of each pixel belonging to the cell boundary (column **(b)**). We obtain this probability from the output of a random forest classifier applied to our rotationally-invariant features (described in Section 5.5) that was previously trained on a set of manually annotated images. We then apply a sequence of watershed transformations at different heights to the probability maps in each section. The outputs of the watershed transformations provide a set of possible 2D segmentations for each section (column **(c)**). We then enumerate a set of possible 3D links that may be used to connect segments in 3D (represented by the arrows in red). Finally, we determine the segmentation fusion of segments and links that identifies each neuron in the stack (column **(d)**).*

fusion framework is flexible and does not require full over-segmentation of each section *a priori*.

We also present a novel set of features for the classification of cellular

structures that provide an accurate, rotationally-invariant summary of connectomic patches. We use this classifier to obtain 2D segmentations for the fusion. An example highlighting the discriminative power of our features is shown in Fig. 5.1. In contrast to previous work, our features do not require hand-tuning of a large set of heuristics or filter banks and efficiently identify cellular structures of different scales and morphologies.

We refer to Section 2.1 for related work on automatic neuron reconstruction.

5.3 Segmentation Fusion for Neural Reconstruction

We provide a high-level picture of our fusion framework in Fig. 5.2. We model each neuron as a group of 2D segments (neuron cross-sections) connected by a sequence of 3D links. We then formulate the problem of identifying the neurons in the stack as the problem of finding the fusion of segments and links that form each neuron.

We start by training a pixel classifier to label individual cellular structures including mitochondria, neurotransmitters, and cellular boundaries in each 2D section. The classifier takes a patch centered around each pixel, computes a compact feature descriptor, and outputs the probability of the center pixel belonging to each class (an example of the labeling is shown in Fig. 5.1). We provide details about the feature descriptor we use for classification in Section 5.5.

Once the pixel classifier has assigned a probability per class to each pixel, we

enumerate a set of possible 2D cross-sections of neurons in each section. To do this, we apply multiple watershed transforms on the probability map for the boundary class obtained on each section (one watershed for each height in the map). Each watershed outputs a 2D partitioning of the section into several possible cell cross-sections. Together, all the watersheds provide a large set of 2D segments that may be used to identify cross-sections of neurons in each section.

We then enumerate a set of 3D links that connect pairs of the 2D segments obtained from the watersheds across consecutive sections. We enumerate those links that connect pairs that spatially overlap in XY (the image plane), but that belong to two different, consecutive sections in Z (see Fig. 5.3).

Once we have a set of candidate segments and links between them, we formulate our fusion problem. As mentioned earlier, we consider each neuron to consist of a sequence of segments (cross-sections) and links, and our goal is to find the fusion that forms each neuron. We formulate the solution to the fusion problem as the maximum a posteriori (MAP) labeling of an MRF, subject to a set of pre-defined clustering constraints. The MRF provides a convenient factorization of the variables in the problem, while the constraints help us to add prior knowledge about the relationship between segments and links. As we explain later, we use constraints to prevent, among other things, the selection of two segments for the final fusion that overlap in the same section (because we know that one pixel can only belong to one cell), and to make sure that, if a 3D link connecting two 2D segments from two consecutive sections is selected, the 2D segments are also selected. We formalize this in the next section.

5.4 Modeling Fusion with MAP-MRF

We start by associating indicator variables (binary variables) s_i and l_j with each 2D segment and 3D link enumerated before. A segment is assumed selected for the final segmentation if its indicator variable is activated (e.g., $s_i = 1$), and similarly for a 3D link (e.g.,, $l_j = 1$). This way, a 3D segmentation of the data is simply specified by a labeling of the indicator variables.

With these definition, we model the solution to the 3D segmentation problem as the MAP selection of segments and links given the image data subject to the fusion constraints. For simplicity, we formulate the posterior probability with the following factorization:

$$P\left(\mathbf{s},\mathbf{l}|\text{data}\right) = \frac{1}{Z}\psi_{\text{SAT}}\left(\mathbf{s},\mathbf{l}\right)\prod_{i,j=1}^{S,L}\psi_s(s_i)\psi_l(l_j), \tag{5.1}$$

where Z is the partition function [153], the vectors $\mathbf{s}$ and $\mathbf{l}$ represent the indicator variables, S and L are the number of 2D segments and 3D links in the enumeration, and $\psi_s\left(s_i\right) = \exp\left(\theta_{s_i}s_i\right)$ and $\psi_l\left(l_j\right) = \exp\left(\theta_{l_j}l_j\right)$, and ψ_{SAT} are compatibility functions of an MRF defined over the indicator variables [153].

The parameters θ_{s_i} and θ_{l_j} measure the relative strength of a 2D segment s_i and a 3D link l_j in the posterior probability, respectively. Dropping the indices i and j for the moment, in our experiments we set each θ_s from the output of our 2D pixel classifier by measuring the strength of the membrane probability on the boundary. As for θ_l, we measure the cross-correlation and displacement between the pair of segments that are connected by the link in question. The more similar the segments are, and the closer they are to each other, the more likely they are

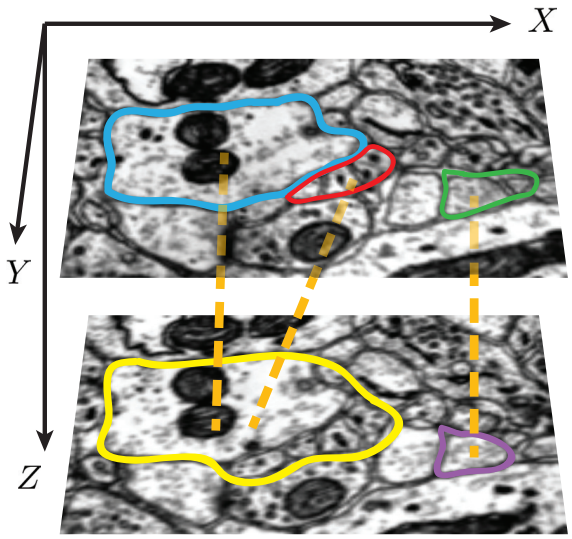


Figure 5.3: *We enumerate links between every pair of segments that belong to two consecutive sections and that overlap in XY. Above, the yellow segment overlaps in XY with the blue and red segments, and not with the green segment.*

presumed to be connected. However, other choices for θ_s and θ_l are possible.

Both θ_s and θ_l should take into account the relative size of the segments being considered. Specifically they should guarantee that, any large segment, aside from the image data, can compete equally against a set of much smaller segments that may be eligible to cover the same 2D area. In our experiments, we choose to achieve this calibration by setting θ_s and θ_l as follows: $\theta_s = \overline{\theta_s}\,\text{size}\,(s)$ and $\theta_l = \overline{\theta_l}\,\text{size}\,(l)$, where $\text{size}\,(s)$ measures the size of a segment (e.g., in pixels), and $\text{size}\,(l) = \text{size}\,(s_a) + \text{size}\,(s_b)$ measures the size of a link connecting two segments s_a and s_b. The parameters $\overline{\theta_s}, \overline{\theta_l} \in [0, 1]$ measure the normalized weight of a segment s and a link l, respectively, as described before.

As mentioned earlier, some fusions of segments and links are physically unrealistic or undesirable given our knowledge about the nature of the problem. We use a special compatibility function, ψ_{SAT}, to give such configurations zero-mass in the MRF (effectively assigning them zero probability of being chosen as a MAP solution), i.e.,:

$$\psi_{\text{SAT}}\left(\mathbf{s}, \mathbf{l}\right) = \begin{cases} 1 : \text{if } \mathbf{s} \text{ and } \mathbf{l} \text{ satisfy fusion conditions} \\ 0 : \text{else.} \end{cases} \tag{5.2}$$

We next enumerate our fusion conditions. The first one prevents undesired solutions, while the second one helps obtain an overall better 3D segmentation. We provide an illustration explaining them in Fig. 5.4.

1. Preventing overlaps between segments:

As indicated earlier, we model each neuron as a group of 2D segments connected by a sequence of 3D links, with each segment representing the cross-section of a neuron. We also know that each pixel can only belong to one neuron at a time. These two facts imply that, for our modeling to be consistent, we cannot tolerate segment overlaps.

One possible way of avoiding overlaps is to require the segmentation to provide a tessellation of the stack (a partitioning without gaps nor overlaps), effectively forcing every region to be covered by exactly one segment. We can achieve this by defining a constraint that requires that $\sum_{i \in \text{o}_\text{k}} s_i = 1$ for every set of overlapping segments o_k. However, in our experiments, we noticed that such constraint can be too restrictive. Depending on the quality of the data, it is sometimes the case that a small 2D region in the stack can only choose from a set

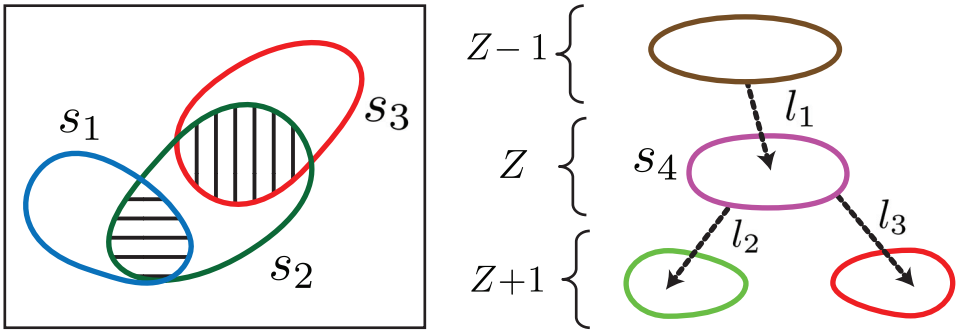


Figure 5.4: ***Left:*** *Three possible neural cross-sections overlapping on the same section. Each of them comes from a different 2D segmentation of the section. Our first fusion condition (see text) requires the MRF to avoid overlaps when deciding on a segmentation for the section. This is enforced by requiring that the indicator variables satisfy* $s_1 + s_2 \leq 1$*, and* $s_2 + s_3 \leq 1$*.* ***Right:*** *Our fusion framework requires the MRF to choose those segments that provide the best continuity in 3D (those that are connected by the best links). The selection of segments depends on the selection of links via our second fusion condition. In this case,* s_4 *must be activated if either* l_1*,* l_2 *or* l_3 *are activated (see Eq. 5.3).*

of segments that may not correspond to anything meaningful in the data. This can cause neighboring regions that may otherwise be able to choose a good segment, be covered by the same wrong choice. To avoid this problem, we choose to relax the previous constraint by simply letting some regions not be covered by any segment, i.e.,, $\sum_{i \in o_k} s_i \leq 1$, which still prevents overlaps between segments.

2. Rewarding good 3D continuity across the stack:

One way of leveraging information from other sections when choosing a segment for a section is to encourage the selection of segments that yield good

3D continuity in the stack. The idea is to make decisions in each section depend on decisions made in the rest of the stack. We achieve this by making the selection of segments and links dependent on each other, and rewarding the selection of segments that are connected by strong links. To do this, we set up the following constraints for every candidate segment s_i:

$$\sum_{j \in \mathrm{TOP_i}} l_j \leq s_i, \quad \sum_{j \in \mathrm{BOT_i}} l_j \leq s_i, \tag{5.3}$$

where $\mathrm{TOP_i}$ specifies the set of links connecting the segment s_i from the top and $\mathrm{BOT_i}$ the ones connecting it from the bottom (in Fig 5.4, $\mathrm{BOT_4}$ for segment s_4 would include the links l_2 and l_4, and $\mathrm{TOP_4}$ the link l_1). To see how this condition works, notice that a link necessarily connects one segment from the top, and one segment from the bottom, so from Eq. 5.3, its activation requires the activation of the corresponding segments.

Solving the segmentation fusion. As stated earlier, we determine the solution to our fusion problem as the MAP assignment to our vectors of the indicator variables $\mathbf{s}$ and $\mathbf{l}$. Such an assignment is obtained by solving $\arg\max_{\mathbf{s},\mathbf{l}} P(\mathbf{s},\mathbf{l}|\text{data})$, with the posterior probability given by Eq. 5.1, which yields the following binary linear programming problem:

$$\begin{aligned} \arg\max_{\mathbf{s},\mathbf{l}} \; & \sum_{i=1}^{S} \theta_{s_i} s_i + \sum_{j=1}^{L} \theta_{l_j} l_j \\ \text{s.t.} \; & s_i, l_j \in \{0,1\}, \\ & \psi_{\mathrm{SAT}}(\mathbf{s},\mathbf{l}) = 1 \end{aligned} \tag{5.4}$$

We solve Eq. 5.4 using a general-purpose binary linear programming solver. We discuss implementation details and running times in Section 5.6.

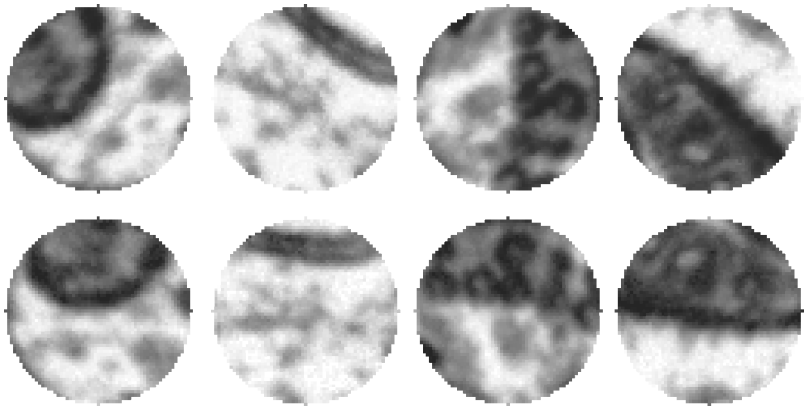


Figure 5.5: ***Top:*** *Exemplar circular patches of brain tissue.* ***Bottom:*** *Inverse discrete Zernike transfrom [154] applied to the rotation normalization of the patches provided by our features. Given a patch, the correct classification of its center pixel is invariant to orientation (but not necessarily to changes in scale or translation). Our features automatically normalize each patch to a referential orientation (in the examples, the disks are automatically rotated to move lower intensities towards the top of the disk). This rotation normalization can help reduce the sample complexity and the size of the training set for the classification of the center pixel of the patch.*

5.5 Computing 2D Pre-segmentations with the Adapted Zernike Features

In this section, we introduce an orthogonal set of rotationally-invariant features for labeling cellular structures in stacks of brain tissue based on the recently introduced Discrete Zernike Transform (DZT) [154]. We use these features to, first, label each pixel as belonging to either mitochondria, cellular boundary,

neurotransmitter, or cellular space (inside and outside) based on a surrounding patch; and second, to enumerate the possible 2D segments (neural cross-sections) that are needed for the fusion.

Rotational invariance is a desired quality when labeling or classifying image patches in connectomics. The 2D orientation of a patch of brain tissue is presumed irrelevant for its classification, and should not condition the labeling or segmentation of the patch, or the labeling of its center pixel (see Fig. 5.5).

The design of rotationally invariant patch descriptors for applications such as texture classification or image categorization is well discussed in the literature [155,156]. However, most of these descriptors also provide invariance to general affine transformations such as scale or translation, which is not necessarily desired in connectomics. For example, the correct label of the center pixel of a patch may depend on which cellular structure the center pixel lies on within the patch (i.e.,, a translation of the patch).

Recently, several authors have addressed the design of features that target specific cellular structures such as mitochondria [28,29], cell boundaries [157], or synapses [158]. Most of these descriptors require the adjustment of a set of parameters depending on the morphology or geometry of the object of interest. Moreover, they often rely on a set of heuristics and filter banks chosen by hand, and are not guaranteed to provide an orthogonal (non-redundant) basis for encoding the original patch.

Our features build upon the DZT [154] to provide a rotationally-invariant and orthogonal descriptor of circular patches. Zernike polynomials have long been

used to build image moments with affine invariance [159], but a common problem in their application has been that they do not form a complete basis on the sampled disk [154]. As a consequence, the computation of moments usually requires redundant disk sampling and least-squares fitting, which leads to numerical difficulties [154].

The DZT avoids these problems, providing an orthonormal basis via non-redundant sampling of the unit disk and an orthogonal factorization of sampled Zernike polynomials [154]. This new disk decomposition offers other important benefits, such as a finite spectrum that allows the design of compact representations of input patches.

In what follows, we show how to build an orthogonal rotationally-invariant feature set from the DZT of an input patch. We refer the reader to the paper by Rafael et al. [154] for specific details about the DZT.

We define the Zernike decomposition of a patch I as:

$$I(\rho, \theta) = \sum_{m,n} c_{m,n} Z_{m,n}(\rho, \theta), \tag{5.5}$$

where $c_{m,n}$ are the coefficients of the DZT, (ρ, θ) are polar coordinates indexing the image patch within the unit disk, and $Z = \{Z_{m,n}\}$ is the set of Zernike polynomials. The integers m and n index the radial and angular frequencies on the unit disk, with $n \in [0, 1, 2, ...]$, $m \in [-n, n]$, $(n - m)$ even, and the total number of harmonics determined by the number of samples on the input disk [154]. For reference, we show the Zernike polynomials $Z_{m,n}$ up to fourth degree in Fig. 5.6.

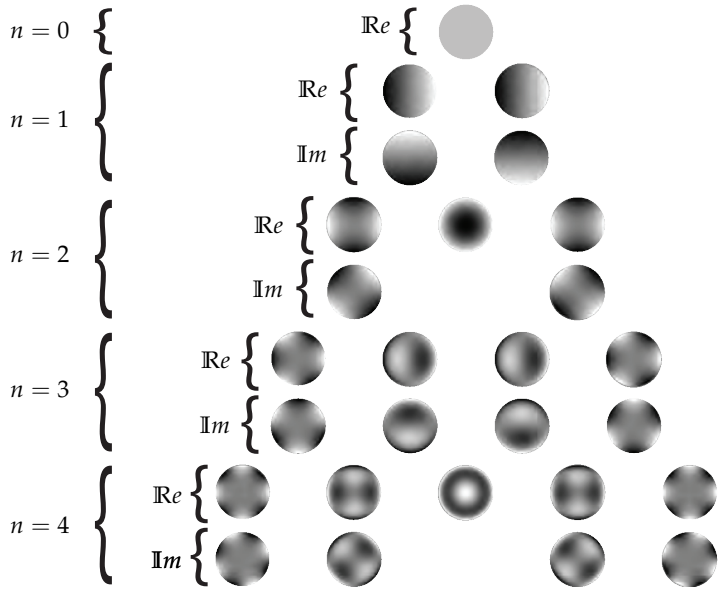


Figure 5.6: *The Zernike polynomials $Z_{m,n}$ up to the fourth radial degree (adapted from [159]). The number of angular components (displayed in columns and indexed by m in Eq. 5.5) is a function of the radial frequency (displayed in rows, and indexed indexed by n in Eq. 5.5). The radial bands are indicated on the left, and the angular bands are displayed in columns. For each radial band n, the angular frequency m ranges between $-n$ and n, with m an integer value subject to the constraint that $n - m$ is even. The real components of each harmonic are shown in the 1st row (n = 0, m = 0), 3rd row (n = 1, m ∈ {-1, 1}), 5th row (n = 2, m ∈ {-2, 0, 2}), 7th row (n = 3, m ∈{ -3, -1, 1, 3}) and 9th row (n = 4, m ∈{-4, -2, 0, 2, 4}). The imaginary components of each harmonic are shown in the 2nd, 4th, 6th, 8th, and 10th rows, respectively.*

The forward and inverse DZT can then be computed as $\mathbf{c} = \mathbf{Q^T i}$ and $\mathbf{i} = \mathbf{Qc}$, respectively, where $\mathbf{c}$ represents the vector of DZT coefficients, $\mathbf{i}$ the vector of input samples, and $\mathbf{Q}$ the orthogonal basis that results from the QR

decomposition of Z.

Grouping each radial band (i.e.,, grouping the harmonics by their corresponding radial frequencies m), we define $\mathrm{DZT_k}$ as the low-pass filtering of the DZT spectrum that keeps only the first k radial bands of the discrete Zernike spectrum. Such filtering builds on the assumption that the low-frequency bands of the DZT spectrum are usually sufficient for classification [159] (see Fig. 5.7 for some reconstruction examples).

Since the phase of the DZT harmonics is linear with respect to rotations of the input patch [159], we can compute the rotation-normalized spectrum $\overline{\mathrm{DZT_k}}$ of $\mathrm{DZT_k}$ by shifting the phase of all of its harmonics by the phase of a low-frequency strong harmonic [159]. This allows us to obtain the same descriptor for rotated instances of the same patch by normalizing with respect to the orientation of large visual cues, which tend to be robust to noise and small differences between patches of the same class. We search for this harmonic by first starting with low-radial and low-angular frequencies and then moving into higher frequencies. There are several search strategies for identifying such harmonics [159], but choosing the first harmonic above a small threshold (50%) worked well in practice. Assuming that $|A_{\hat{m},\hat{n}}|e^{j\hat{m}\phi_{\hat{m},\hat{n}}}$ is a strong normalizing harmonic (in polar form), we normalize each harmonic $A_{m,n}$ in $\mathrm{DZT_k}$ as $\overline{A_{m,n}} = A_{m,n}e^{-jm\phi_{\hat{m},\hat{n}}}$.

Note that the QR decomposition is only computed once for all patches (i.e.,, not for every patch). This way, we only need to compute $\mathbf{c} = \mathbf{Q^T i}$, prune the harmonics beyond a radial band k defined *a priori*, and apply the rotation

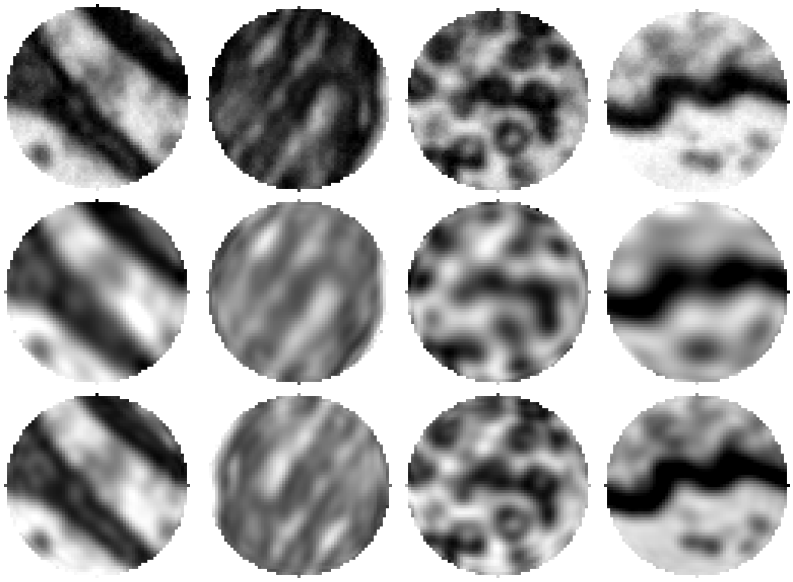


Figure 5.7: *Disk reconstructions from the inverse discrete Zernike transform after low-pass filtering,* $\mathrm{DZT_k}$*, for different values of* k*.* ***First row:*** *Original disks.* ***Second row:*** *Reconstructions for* $k = 10$*.* ***Third row:*** *Reconstructions for* $k = 15$*. Our Adapted Zernike Features retain the large visual cues from the original patch within only a few bands. The highest possible value of* k *in all these images is 73.*

normalization to obtain our descriptor for a given patch $\mathbf{i}$. In Section 5.6, we comment on the selection of the radial band k that we used for our experiments.

5.6 Experimental Results

In this section, we provide experimental evaluation of our fusion framework and our feature descriptor. We first evaluate our fusion method against previous solutions for neuron segmentation and general segment clustering. We then

compare separately our Adapted Zernike Features against other connectomic-specific features and more general rotationally-invariant patch descriptors.

For all our tests we used four ssEM stacks of brain tissue from the somatosensory cortex of an adult mouse. Each stack is 1K x 1K pixels x 9 sections deep. The sections have an estimated thickness of 20 nm, and each pixel has an estimated physical dimension of 4 nm x 4 nm.

Evaluation of Our Fusion Framework. We use IBM's CPLEX [1] for solving the binary integer programming problem of Eq. 5.4. In our experiments, we had an average of 20K variables per volume evaluated, and the solver took barely 5 seconds to determine the solution on a desktop PC.

We compare our fusion framework with two recently developed segment-clustering methods for connectomics: LP-R, a method for image co-clustering based on linear programming relaxation [31], and the agglomerative clustering approach employed in [26], which we refer to as AC. We also test against MHVS. Finally, as a baseline and to evaluate the benefit of working with multiple segmentations, we include the results from running our fusion framework when only one 2D segmentation per section is provided (which we refer to as Fusion-1). For this last comparison, we obtain the 2D segmentation for each section by labeling each pixel with our pixel classifier.

In order to give all the methods the same initial advantage, we used the same pixel classifier (ours) to provide the initial 2D pre-segmentations. For those

[1] ibm.com/software/integration/optimization/cplex-optimizer/

methods that required an initial 2D over-segmentation (e.g. LP-R and AC), we run our pixel classifier at higher boundary detection rates until every cell was initially over-segmented.

Fig. 5.8 shows the average Rand error for each method when compared with a set of 3D cell skeletons provided by an expert neuroscientist. The Rand error $R(S,H) \in [0,1]$ measures the discrepancy between the grouping of pixels made by the method S and in the ground truth H and is defined as [160]:

$$R\left(S,H\right) = \frac{\text{\# merge errors} + \text{\# split errors}}{\text{\# total pairs}}, \tag{5.6}$$

where a merge error refers to a pair of pixels that were grouped by S, but were not grouped in the ground truth H, and the split error is defined conversely. Since not every pixel was annotated in the ground truth, we only evaluated the Rand error on the pairs of pixels corresponding to the 3D skeletons.

Our fusion framework achieved the lowest error rate in our tests, with errors resulting mostly from false splits (i.e.,, loss of continuity) along the Z axis and the creation of spurious small segments in areas between cells. The majority of neural cross-sections were reconstructured correctly, as we show in the examples in Figs. 5.11 and 5.12.

Evaluation of our Features. We compare our Adapted Zernike Features with two connectomics-specific descriptors, the Ray features [29] and the Radon-like features [28]. In contrast to our features, these descriptors were designed to classify specific types of cellular structures such as mitochondria [29] and require adjusting several parameters depending on the structure of interest [28]. Since our features were designed with the goal of providing a robust rotation-invariant

descriptor, we also test against more general rotation-invariant descriptors and filter banks such as MR8 [155], and the feature set LBP-HF [156].

We provide the precision-recall curves from the pixel labeling of the classes mitochondria, cellular boundaries, vesicles, and cellular space in Fig. 5.9. In all comparisons, we used a random forest classifier with 300 trees [161]. A precision-recall curve measures how precision and recall vary as the detection rate (also known as the "regime") for a given class changes. Precision is defined as the fraction of correct (pixel) positives of all the positives detected, i.e., Precision $= \frac{\text{TP}}{\text{TP+FP}}$. Recall is defined as the fraction of positives detected from all the positives in the ground truth, i.e., Recall $= \frac{\text{TP}}{\text{TP+FN}}$, where TP and FP stand for the number of true and false positives, respectively, while TN and FN stand for the number of true and false negatives, respectively. When using a random forest classifier the detection rate (TP + FP) for a given class increases as we lower the threshold to detect pixels from the output of the classifier (the # of votes from the forest toward a given class). This way, we can obtain different detection rates (and therefore different points in the precision-recall curve) by iteratively changing the threshold on the number of votes toward each class.

For training, we used expert pixel annotations on the first four images of each stack. We tested on the rest of images in the stack. For our adapted Zernike features, we used the first 18 radial bands of the DZT spectrum and an input disk of 60 pixels in diameter after downsampling each section by a factor of two. In our tests, computing the feature descriptor and classifying every pixel of each testing stack takes less than 3 hours if run on a modern PC, and less than 10 min.

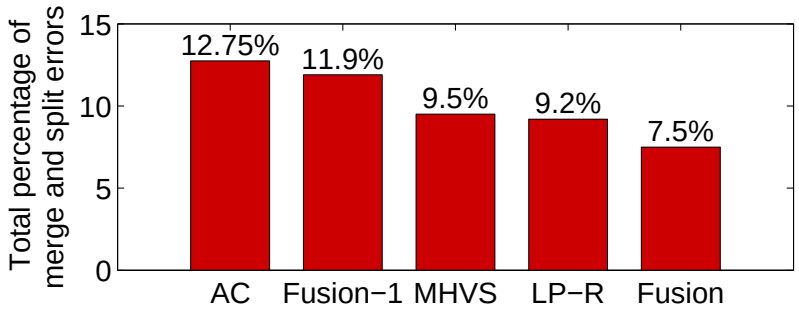


Figure 5.8: *Average percentage of merge and split errors (i.e., average Rand error) in four ssSEM stacks with respect to the ground truth provided by an expert neuroscientist in the form of cell skeletons. Fusion outperforms other state-of-the-art methods demonstrating the benefit of explicitly enumerating possible segments and links for the final segmentation over solutions based on region clustering.*

if run on a grid with 25 machines. For MR8 and LBP-HF, we used the image patch that encloses such disk. For the other features, we used the parameters originally reported by the corresponding authors. In all our tests, our adapted Zernike features outperformed the competing descriptors. We show additional examples of the automatic pixel labeling of mitochondria, cellular boundaries, vesicles, and cellular space with our features in Fig. 5.10.

5.7 Summary

In this chapter, we addressed the problem of automatic 3D segmentation of ssEM image stacks. We presented a framework that leverages information from multiple sections and considers several segmentation choices in each section to

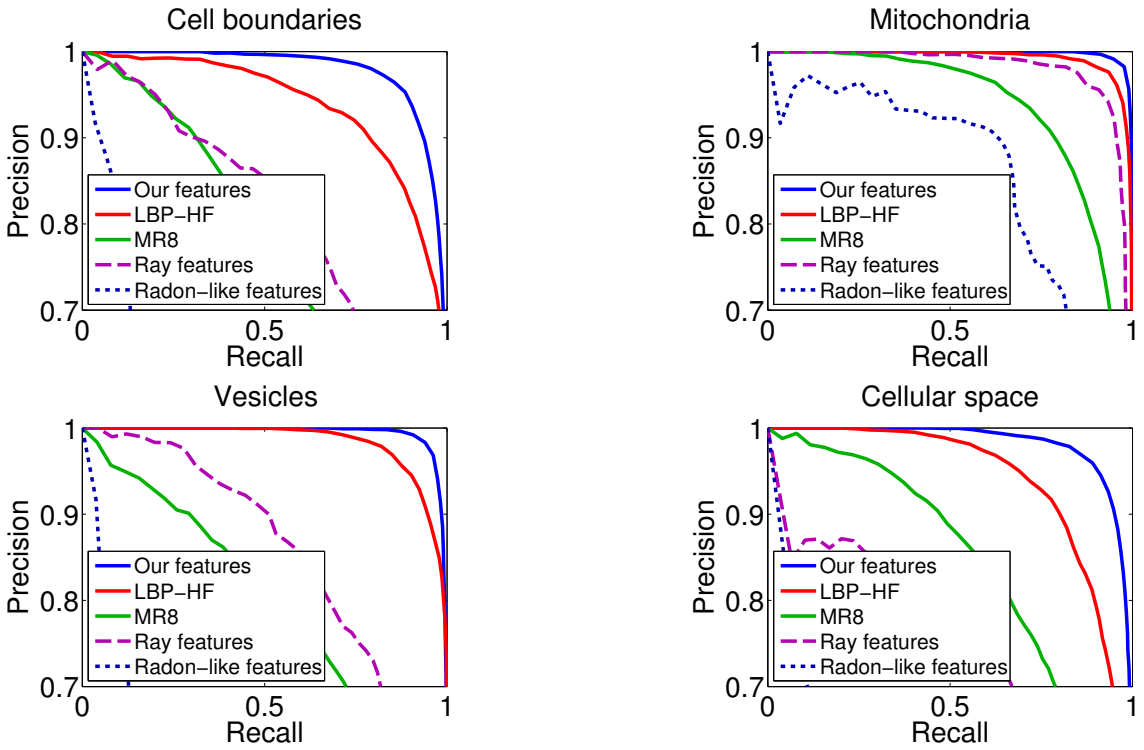


Figure 5.9: *Precision and recall curves for our Adapted Zernike Features and other competing features when labeling cellular structures. Our features are highly discriminative even when dealing with structures of different scales and morphologies, outperforming all the other descriptors on every class evaluated.*

determine the final global partitioning. Unlike previous efforts that perform segmentation by clustering regions from an initial oversegmentation, our method is able to directly evaluate candidate segments that may be used for the final partitioning of each section.

The framework described in this chapter does not handle neural mergers and splits, but we are currently exploring several strategies to cope with them. We

discuss one of these strategies in Chapter 6.

Finally, we have also presented a highly discriminative set of rotationally-invariant features for connectomics. Our features can target cellular structures of different scales and morphologies and only require adjusting two parameters; the size of the input disk, and the number of radial bands in the DZT. Together, and individually, our features and fusion method gave the best segmentation and reconstruction results.

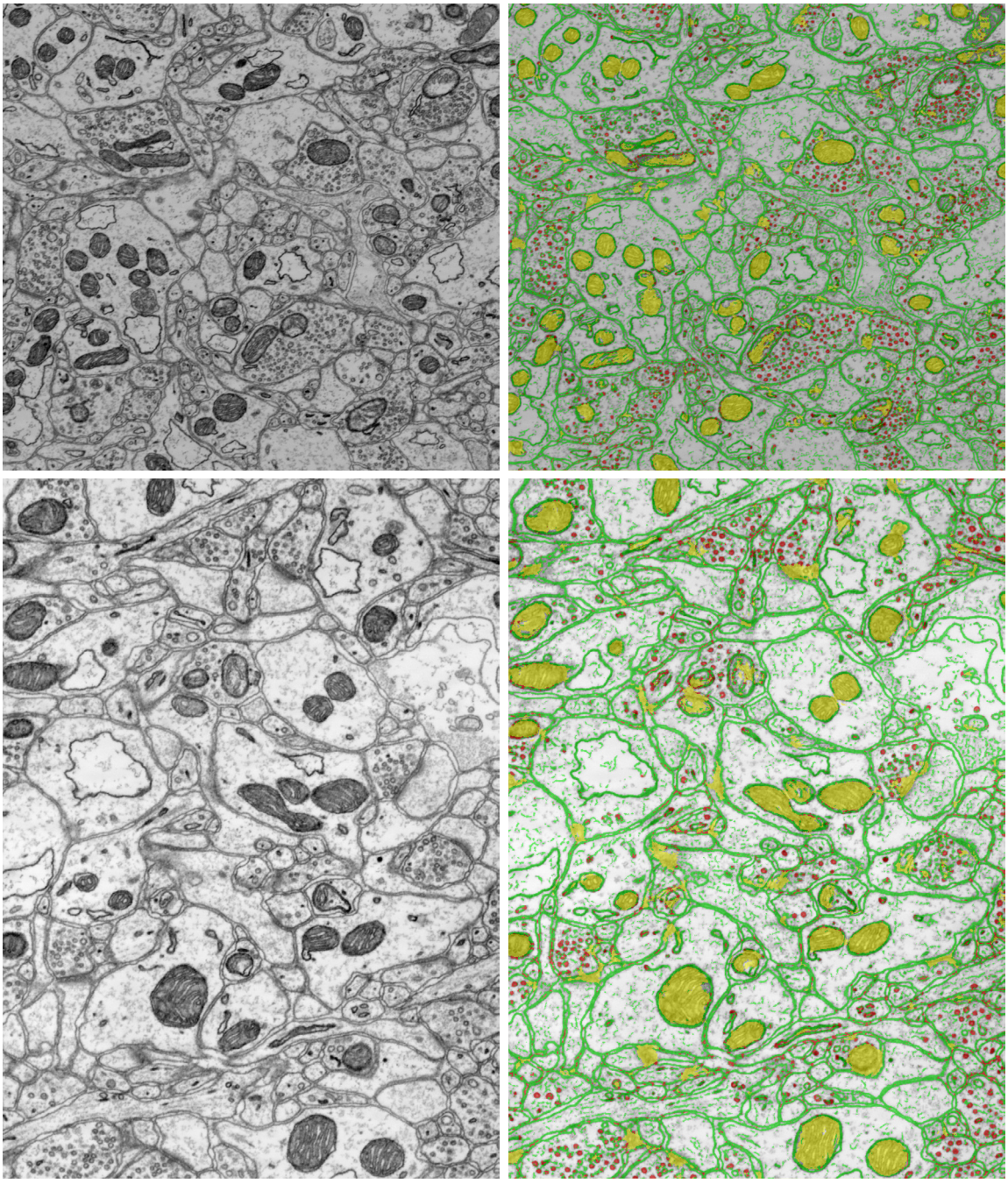


Figure 5.10: *Additional examples of the automatic labeling (right) of cellular structures of 2D sections of brain tissue (left) using our features. Each pixel is labeled by a random forest classifier trained with the features we introduce in Section 5.5. Mitochondria are shown in yellow, vesicles in red and cellular boundaries in green. The rest of pixels are labeled as cellular background (shown with a transparent label).*

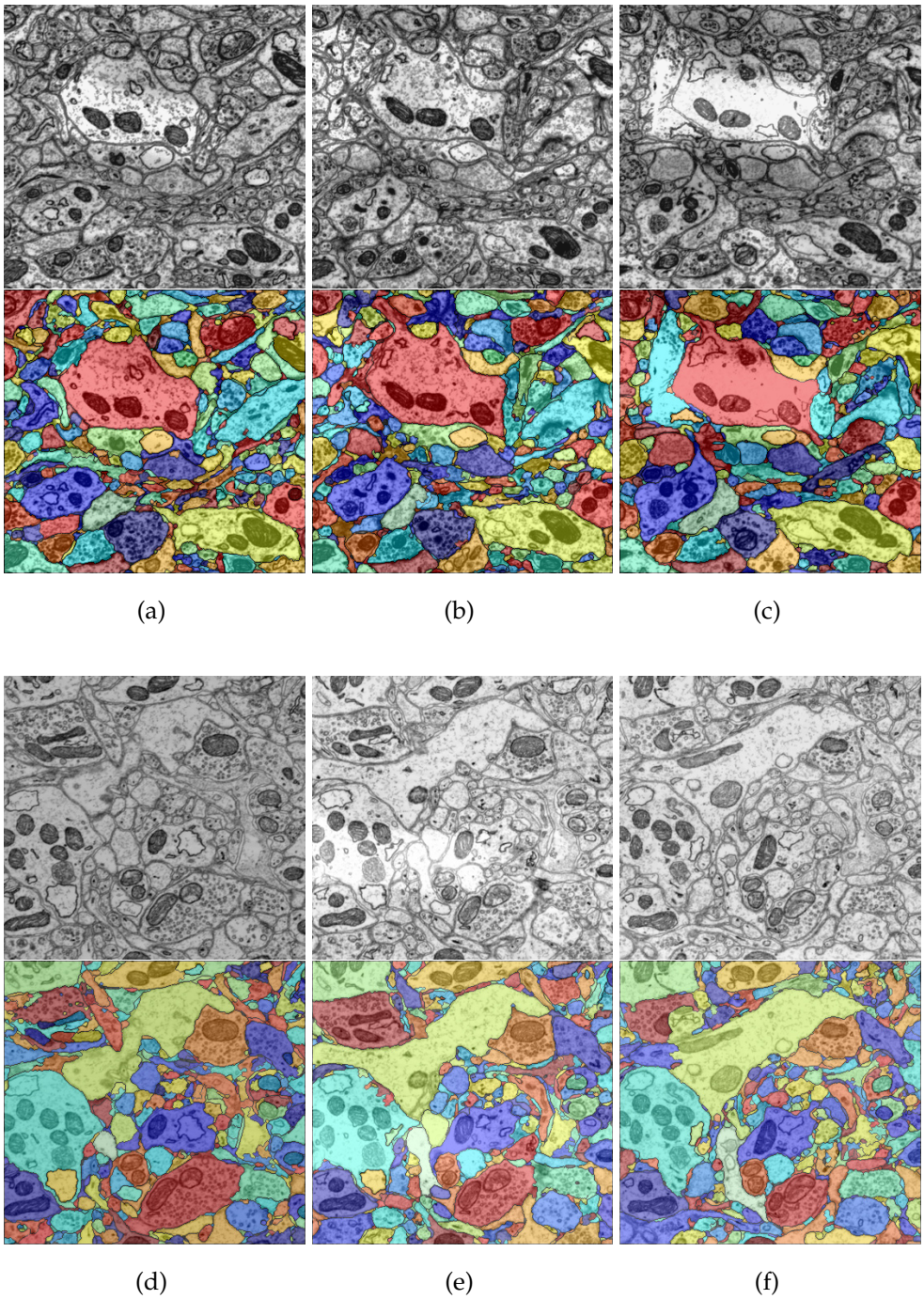
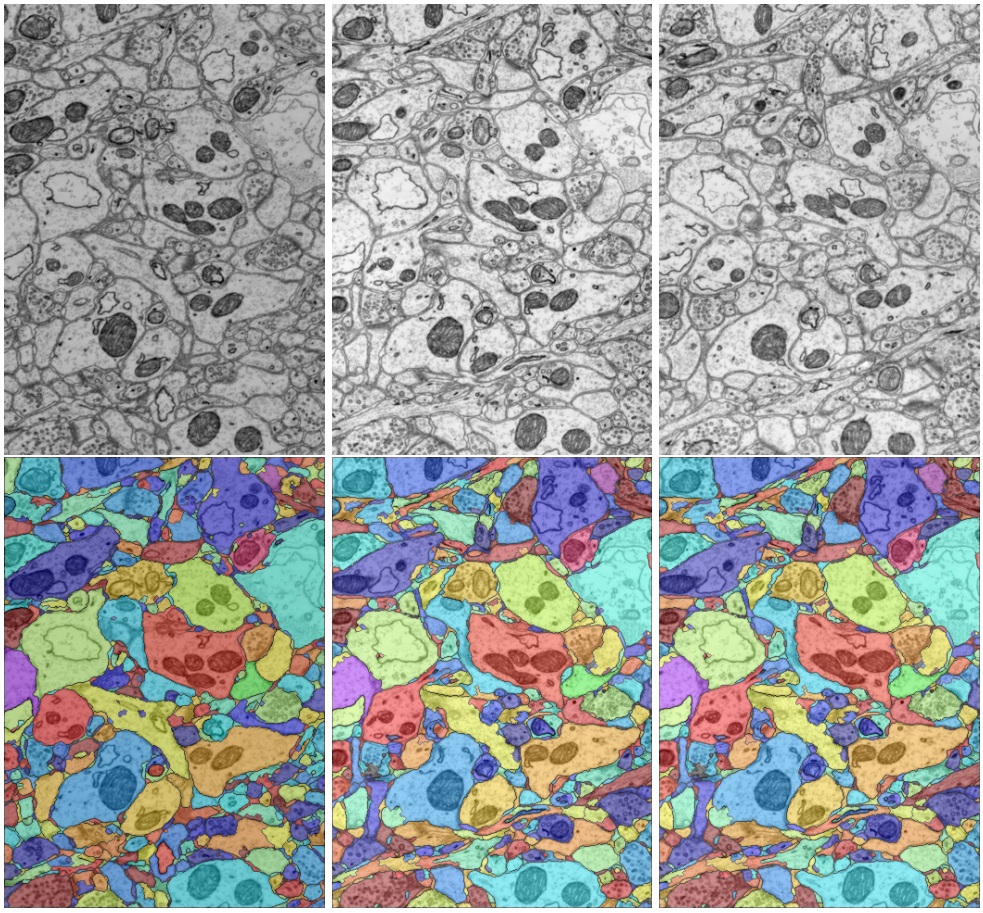
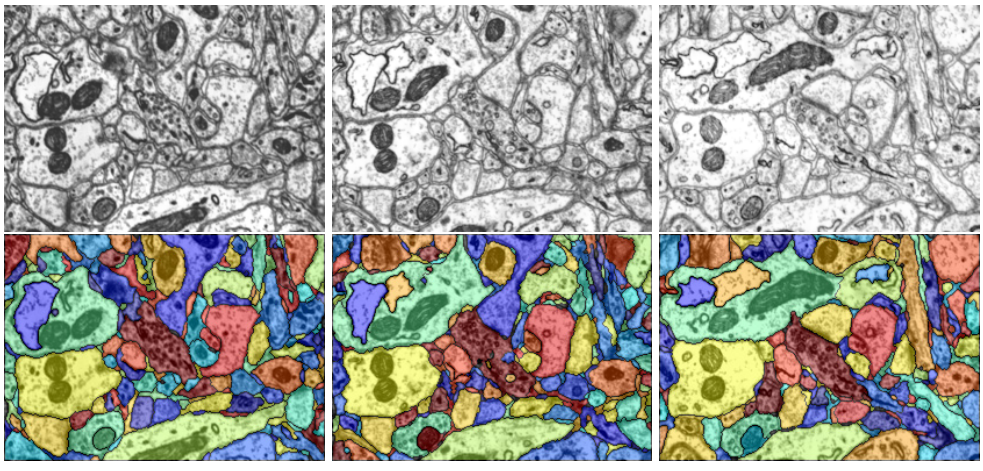


Figure 5.11: *Results of Segmentation Fusion. The images (a), (b) and (c) come from one stack, while the images (c),(d) and (e) come from a different stack. The images correspond to sections 1, 5 and 9 from each stack. Our fusion framework correctly grouped most of the neurons in the stack, even in highly saturated regions of the images.*



(a) (b) (c)



(d) (e) (f)

Figure 5.12: *More results of Segmentation Fusion. The images (b), (b) and (c) come from one stack, while the images (c),(d) and (e) come from a different stack. The images correspond to sections 1, 5 and 9 from each stack.*

CHAPTER 6

Scaling Strategies and Enabling Branching in Segmentation Fusion

As part of ongoing efforts, in this chapter we discuss two extensions to Segmentation Fusion, the framework described in the previous chapter. We propose (1) divide-and-conquer schemes that are amenable to parallelization and that allow Fusion to segment relatively large image stacks, and (2) an extension to the original Fusion optimization problem of Section 5.4 that enables Fusion to connect individual 2D segments with multiple 2D segments from adjacent sections (we call this "branching"). We include preliminary results of these extensions and evaluate the results against ground truth collected by trained neuroscientists. We also show what we think are the largest automatic neuron reconstruction results ever obtained with ssSEM.

6.1 Scaling Segmentation Fusion

For scaling Fusion, we propose two divide-and-conquer schemes, *Bipartite Fusion* and *Nested Fusion*. We show an overview of Bipartite Fusion in Fig. 6.1. The first step is to divide a large data volume into a set of smaller subvolumes of approximately the same size in pixels. We then obtain a 3D segmentation for each of these sub-volumes independently using Fusion, the method described in the previous chapter. Once we have a 3D segmentation of each subvolume, we solve a bipartite matching problem between 3D segments for every pair of adjacent subvolumes. To do the matching, we formulate a linear two dimensional assignment problem, also known as a maximum weighted bipartite matching problem [162], on the adjacency graph between 3D segments from each pair of subvolumes. The result of these matchings is a large graph that tells us how the 3D segments in the subvolumes connect throughout the entire stack. The connected component labeling of this graph yields the final 3D grouping of pixels or segmentation of the full volume.

When formulating the maximum weighted bipartite matching problems, we define the cost of establishing a link between a pair of 3D segments from adjacent subvolumes using the same parameter based on shape similarity, θ_l, that we used in Section 5.4, that is, the dice coefficient of the binary masks of the segments. For solving each weighted bipartite matching problem, the Edmonds-Karp algorithms or the work of [163] with run time complexity of $\mathcal{O}\left(n^2 \log n + nm\right)$ (n and m being the number of segments on each side of the graph) provide some the most efficient algorithms. In practice, however, one can also use a general integer

programming solver such as CPLEX to solve this problem for every pair of adjacent volumes, since the number of total 3D segments to match is typically small ($\sim$1000). We note that if n and m are not equal, solutions involving a perfect matching (i.e., involving all segments) are not feasible in a bipartite matching problem.

A variation of *Bipartite Fusion* is *Nested Fusion*. The idea is similar, but instead of fixing a single 3D segmentation per subvolume, we obtain multiple 3D pre-segmentations from Fusion for each of them (in practice, solvers such as CPLEX can be asked to output multiple close-to-optimal solutions for a given optimization problem). Then, instead of solving a bipartite matching problem between the 3D segments of the segmentations of the adjacent subvolumes as in Bipartite Fusion, we formulate a new Fusion problem (hence the name *Nested Fusion*). This new Fusion problem is defined to yield the set of 3D segments from the pre-segmentations of each subvolume, and the set of links between these 3D segments that give the most globally consistent segmentation. The formulation of Nested Fusion is exactly the same as the original formulation for Fusion in Section 5.4, but with the variables s_i associated to 3D segments (as opposed to 2D segments) and the variables l_j associated with links between 3D segments across cubes (as opposed to links between 2D segments between sections). We show a pipeline of this scheme in Fig. 6.3.

Figure 6.1: *For segmenting large datasets we propose a divide-and-conquer approach that we call Bipartite Fusion.* ***(a):*** *First we divide the full volume into smaller subvolumes of approximately the same size.* ***(b):*** *We then segment each subvolume independently in 3D using Segmentation Fusion, the method we described in the previous chapter.* ***(c)****: We then solve a bipartite matching problem (a linear assignment problem) between every pair of adjacent cubes. The result of the matchings is a graph that connects the 3D segments across cubes. The connected component labeling of this graph outputs the consistent labeling of the full stack.* ***(d)****: We finally stitch back the results across cubes to obtain the segmentation of the full stack.*

6.2 Coping with Splits and Mergers in Fusion

For enabling Fusion to cope with branching (the splitting of neural processes across sections) we propose replacing the Fusion constraint of Eq. 5.3 with a different constraint, and the use of a heuristic to prevent overmerging.

Recall that the original expression for the constraints of Eq. 5.3 originally was:

$$\sum_{j \in \mathrm{TOP_i}} l_j \leq s_i, \quad \sum_{j \in \mathrm{BOT_i}} l_j \leq s_i \tag{6.1}$$

where $\mathrm{TOP_i}$ specified the set of links hitting the segment s_i from the top and $\mathrm{BOT_i}$ the ones hitting it from the bottom. Since the variables s_i and l_j are binary, the allow constraints serve two purposes. First, they couple the activation of links l_j hitting a segment s_i with the activation of the segment, so that if any link

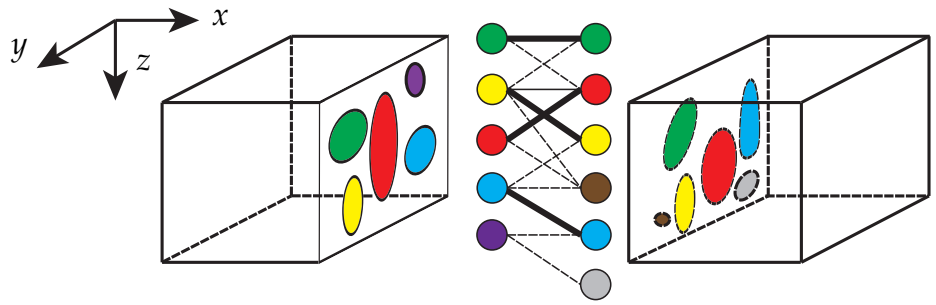


Figure 6.2: *In Bipartite Fusion, we solve a bipartite matching problem between 2D segments from pairs of adjacent 3D subvolumes. This figure illustrates the matching for two subvolumes adjacent in the axis x. The 2D segments on each subvolume in color are 2D sections of 3D segments obtained from the 3D segmentation of each subvolume (with Fusion). We obtain the costs for the bipartite matching problem (associated with the weights of the edges in the graph illustrated above) in the same way we did in Section 5.4; i.e., we compute a cost value θ_{l_j} for each link j between a pair of 2D segments as the cross-correlation of the masks of the segments that are connected by the link.*

l_j is activated in any of the equations above, the associated segment $s_i$is activated as well. This effectively makes the selection of segments in each section of stack depend on the selections made on other sections. To see this note that the activation of links between two sections conditions the activation of links on the consecutive sections. And second, they constrain the number of links that can be activated in TOP_i and BOT_i to be one.

Consider replacing the constrain above by:

$$\sum_{j \in \text{TOP}_\text{i}} l_j \leq |\text{TOP}_\text{i}|\, s_i, \quad \sum_{j \in \text{BOT}_\text{i}} l_j \leq |\text{BOT}_\text{i}| s_i \tag{6.2}$$

where $|\text{TOP}_\text{i}|$ and $|\text{BOT}_\text{i}|$ are the number of links hitting the segment s_i from the

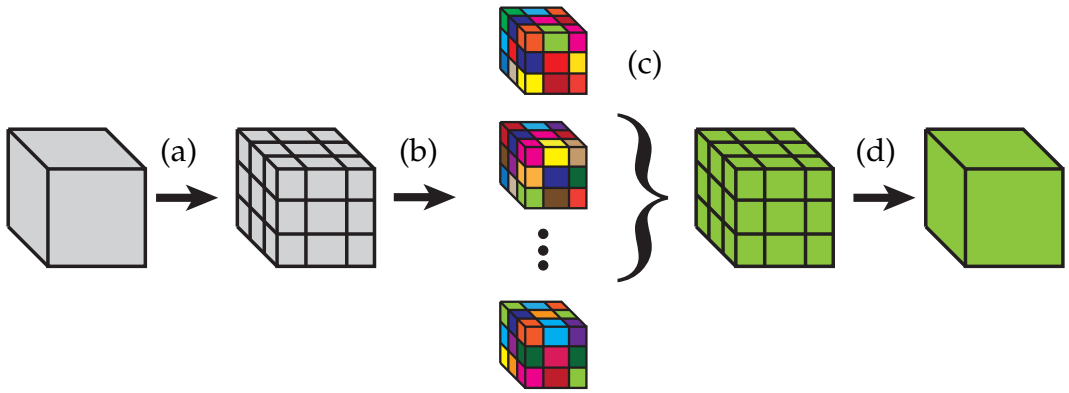


Figure 6.3: *An alternative divide-and-conquer segmentation scheme to Bipartite Fusion is Nested Fusion.* ***(a):*** *We first divide the full volume into smaller subvolumes of approximately the same size.* ***(b):*** *We then segment each subvolume independently in 3D using Segmentation Fusion, the method we described in the previous chapter, obtaining multiple solutions (multiple 3D segmentations) from the solver. We then solve a Nested Fusion problem that selects the set of 3D segments within each subvolume and links between them (across volumes) that yields the most globally consistent segmentation of the full stack. The steps* ***(c)****, and* ***(d)*** *are identical to those described in Fig. 6.1.*

the sections immediately above and below respectively. The new constraint allows for the activation of multiple links hitting any segment either from the sections immediately above or below while still coupling the decision between segments across sections (a segment s_i would be activated if any link hitting it is also activated, which itself conditions the selection of links on subsequent sections).

However, if we allow segments to connect with any number of overlapping segments from adjacent sections and also keep the parameters θ_{l_j} associated with

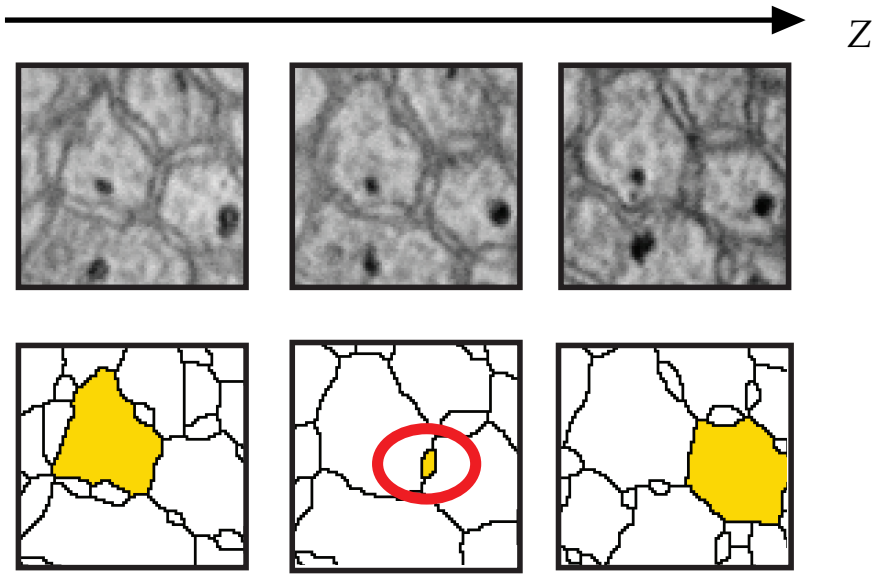


Figure 6.4: *The Equation 6.2 allows Fusion to activate multiple links per segment, which may lead to overmerging. The figure illustrates a case where the yellow segment in the middle section overlaps with the two other yellow segments from the first and last section by just a few pixels. Small segments like the one in the middle are common to appear from the space between cells or from artifacts in pre-segmentations. If the cost associated with the links connecting the first and third yellow segments with the second yellow segment are positive, the maximization of Eq. 5.4 will activate the links between the yellow segments, "chaining" the first and last yellow segments, and merging the two neural processes.*

each link in the cost function positive, the maximization of Eq. 5.4 in the cost function can lead to overmerging (see Figs. 6.4 and 6.5). To prevent this, we let some θ_{l_j} be negative, so that their local activation incurs cost in the optimization. In practice, this is equivalent to forcing such links to not be activated (i.e.,

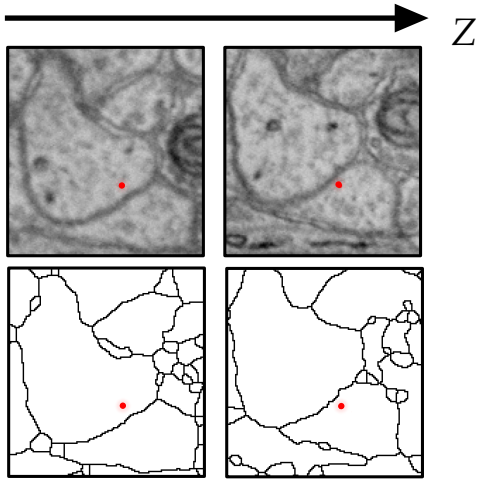


Figure 6.5: *The Equation 6.2 allows Fusion to activate multiple links per segment, which may lead to overmerging. For example, large displacements between sections due to the anisotropy of the stack may lead Fusion to activate a link between two segments that overlap pixelwise but that do not correspond to the same cell (the red dot corresponds to the same (x, y) coordinates on two consecutive sections).*

pruning them), since there is nothing in the original formulation of Eq. 5.4 that would compensate for the cost incurred when activating a link with negative cost. In our experiments we used the following heuristic to determine which links between pairs of segments to prune. Assuming two segments A and B on two consecutive sections, we compute two numbers m and M for the link connecting them:

$$\begin{aligned} m &= \min\{F_{A\to B}, F_{B\to A}\} \\ M &= \max\{F_{A\to B}, F_{B\to A}\}, \end{aligned} \tag{6.3}$$

where $F_{B\to A}$ $F_{A\to B}$ represent the fraction of pixels that "flow" from A to B and

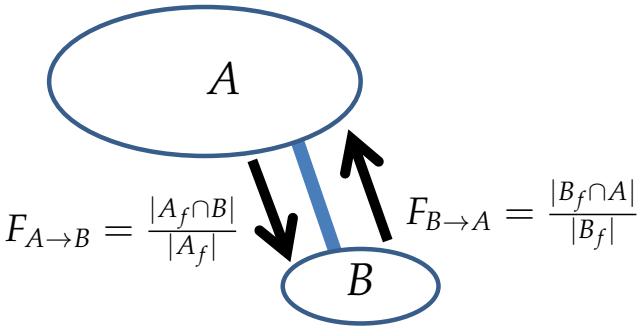


Figure 6.6: *To avoid overmerging when using the extension of Fusion that enables it to cope with branches (i.e., to activate more than one link per segment), for every link between a pair of segments on consecutive sections we measure what fraction of each segment "flows" into the other segment according to optical flow. We then use a heuristic that prunes links that connect segments that barely overlap as given by the maximum and minimum value of these measures (see text). A promising direction for future research may instead try to learn the cost associated with these links from image statistics.*

from B to A respectively and are defined as $F_{A\to B} = \frac{|A_f \cap B|}{|A_f|}$ and $F_{B\to A} = \frac{|B_f \cap A|}{|B_f|}$, where A_f is the projection of A on the section of B according to the optical flow between both sections, and B_f is defined analogously. In practice, we have noticed that pruning links for which $m \leq 5\%$ and for which $M \leq 85\%$ gives good results. This has the effect of pruning the links where the segments connected by the links do not "flow enough" into each other. In our experiments, we used the optical flow method of [164] for computing $F_{A\to B}$ and $F_{B\to A}$.

6.3 Preliminary Results

To test the extensions we proposed in this chapter, we run our methods on two datasets that are much larger than the ones we used in Chapter 5. The first dataset, called **AC3**, is a stack that is 1024 x 1024 x 150 voxels in size. The second dataset, called **5K**, is 5,000 x 5,000 x 1000 voxels in size. We note that if we were to solve the original Fusion formulation directly on this last volume, **5K**, a back-of-the-envelope calculation tells us that we would need to optimize Eq. 5.4 on hundreds of millions of variables. In our experiments, IBM CPLEX (the solver we use in practice) already runs out of memory when working with more than 500,000 variables on a regular desktop workstation.

For the dataset **AC3**, we used the manual segmentation of Daniel Berger, a trained neuroscientist, as the ground truth for comparisons. Daniel reconstructed all the neural processes in this stack manually during a period of two months using an interactive image editor. Examples of this dataset and the manual annotations can be seen in Fig. 1.2 in Chapter 1.

For the dataset **5K**, we used the manual annotations collected by several trained neuroscientists as the ground truth segmentation (courtesy of Bobby Kashthuri, Daniel Berger and other students in our lab). Given the size of this dataset, only a few neural processes were annotated. To get a sense of the size of the ground truth as a fraction of the full dataset, we show renderings of the manual reconstruction in Fig. 6.13

In all these tests we used the method of [30] to obtain the 2D pre-segmentations per section, although other 2D segmentations methods such as the ones we

discussed in Chapter 5 can be used. In total we conducted the following tests:

Experiment A (5K): We test Bipartite Fusion on the dataset **5K** after dividing the original stack into subvolumes of size 5K x 5K x 2 sections (i.e., we divide the original volume into subvolumes only along Z, the imaging plane, every two sections). The idea behind this experiment is to test Bipartite Fusion along only one axis, Z (the axis orthogonal to the acquisition plane), since the image resolution is much lower in Z than in X or Y. We divide every two sections to keep the memory footprint low in each subvolume. Dividing the original stack according to this partitioning scheme results in 1 x 1 x 999 = 999 image subvolumes.

Experiments B and C (5K): We test Bipartite Fusion (B) and Nested Fusion (C) on **5K** using a different partitioning scheme. We test partitioning the full volume along all three axes, in subvolumes of size 1,280 x 1,280 x 15 voxels. This gives each subvolume 15 sections for segmentation and much more context in Z in comparison to Experiment A, where each subvolume had only two sections. The compensate for the extra sections in each subvolume, we reduce the XY dimensions of each subvolume to 1,280 x 1,280 pixels to keep a low memory footprint. This division scheme resulted in 4 x 4 x 71 = 1136 image subvolumes.

Experiments D and E (5K): We test the same configurations as in Experiment B and C, but also giving Fusion the ability to cope with branching using the extension discussed earlier in this chapter.

Experiments F, G, H and I (AC3): We test Bipartite Fusion with and without branching (F and G) and Nested Fusion with and without branching (H and I) on the dataset **AC3**. We divide AC3 into subvolumes of 1,280 x 1,280 x 15 voxels (the same size as in Experiments B,C,D and E).

To get a quantitative sense of the quality of the reconstructions we compare the result of the automatic reconstructions with the manual annotations in each dataset. We used two well-established measures in the clustering literature, the *rand error* [160] that we introduced in Eq. 5.6 in Section 5.6, and the more recent measure *variation of information* [165] that we describe next.

The Variation of Information provides a metric on the space of clusterings [165] and is defined for n points (n pixels in our case) as:

$$\mathrm{VI}\left(S, S\right) = H\left(S\right) + H\left(S'\right) - 2I\left(S, S'\right), \tag{6.4}$$

where $\mathrm{VI}\left(S, S'\right) \in [0, \log n)$, with lower values corresponding to lower error, and H and I represent respectively the entropies and mutual information between two clusterings, S and S', and are defined as:

$$\begin{aligned} H\left(S\right) &= -\sum_{k=1}^{K} P\left(k\right) \log P\left(k\right) \\ I\left(S, S'\right) &= \sum_{k=1}^{K} \sum_{k'=1}^{K'} P\left(k, k'\right) \log \frac{P\left(k, k'\right)}{P\left(k\right) P'\left(k'\right)}, \end{aligned} \tag{6.5}$$

with K and K' the number of clusters in S and S' respectively, and $P\left(k\right)$ and $P\left(k, k'\right)$ measure the fraction of pixels in the cluster k, and on the intersection between two clusters k and k' from S and S' respectively.

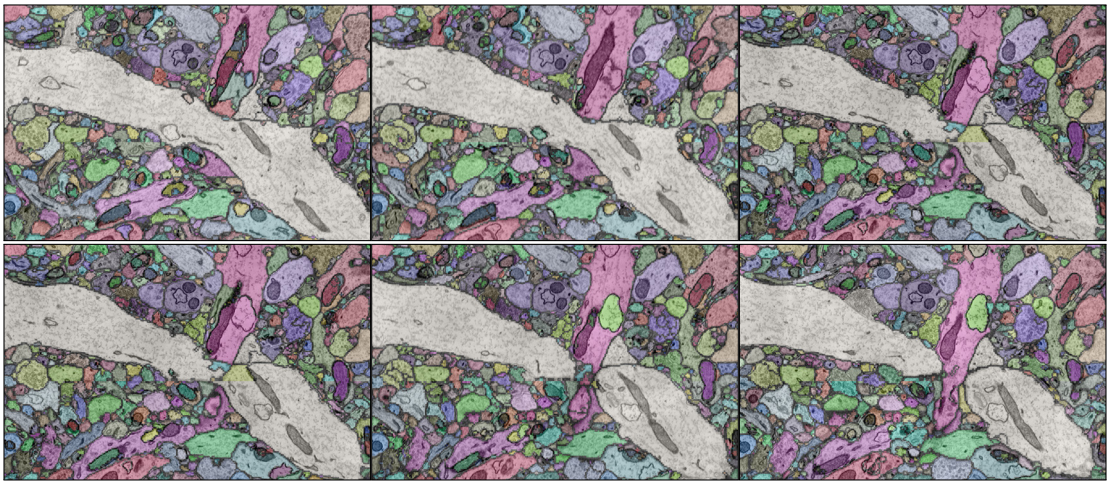


Figure 6.7: ***Left to right, Top to bottom****: Example of Fusion with the proposed extensions handling the splitting of a neural process (the one in brown) across consecutive sections.*

The difference between how the Rand Error and Variation of Information compare two segmentations (two clusterings) can be understood in the way they weight merge and split errors as a function on the difference in cluster sizes [166]. However the perceptual meaning of this difference remains unclear [21]. Moreover the question of how to measure the quality of an automatic neuron reconstruction is active topic of research [25, 167] and we think is a good direction for future work.

We show examples of the reconstruction results that illustrate the ability of Fusion to handle branches in these experiments in Figures 6.7, 6.8, and 6.9. We also show a failure example in Fig. 6.10.

Figure 6.14 shows a 3D rendering of the result for the winning experiment (Experiment C) which obtained the lowest Rand error and lowest Variation of

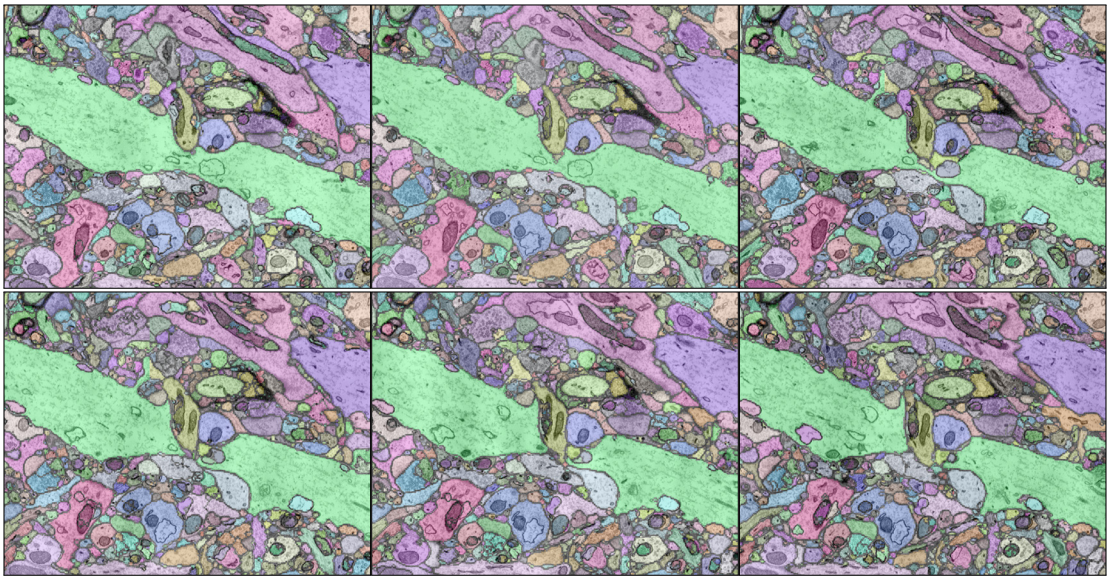


Figure 6.8: ***Left to right, Top to bottom****: Example of Fusion handling the splitting of a neural process (the one in green) across consecutive sections.*

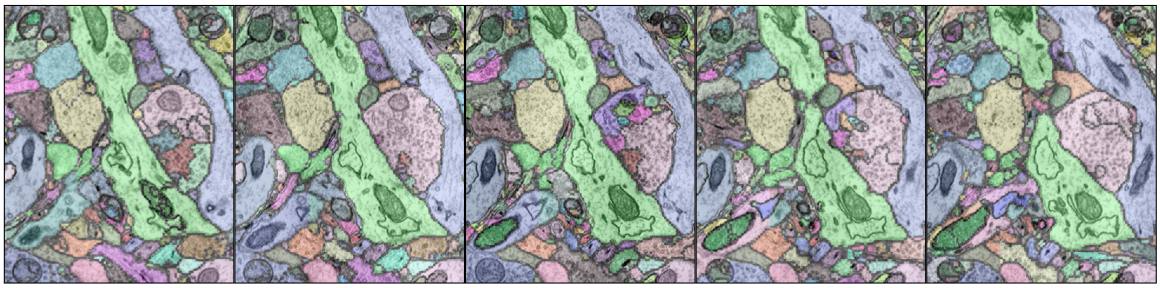


Figure 6.9: ***Left to right****: Fusion handling the splitting of a neural process (the one in green) across consecutive sections.*

Information in the dataset **5K**. We also show cross-sections of this winning result in Figures 6.15 and 6.16. The sections displayed in these two last figures are 1, 20, 250, 300, 500, 550, 800 and 850 out of the 1000 sections in the volume. We show

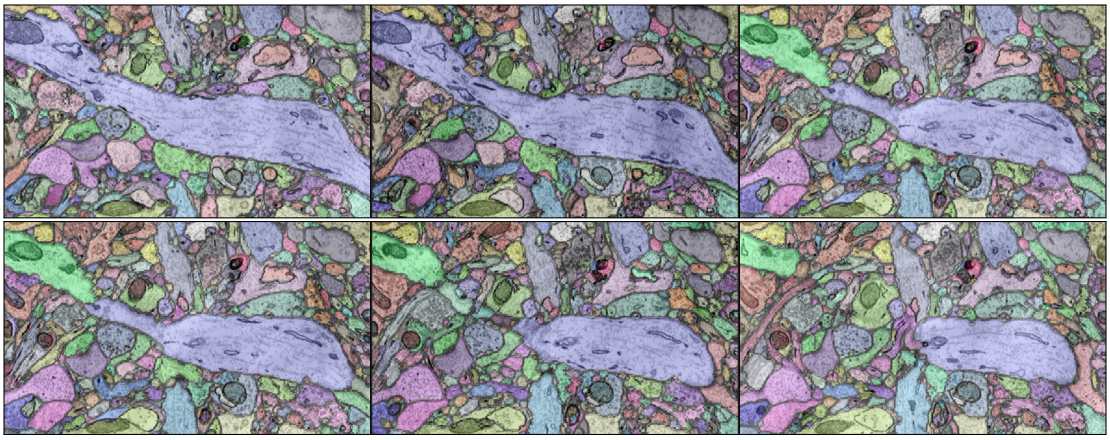


Figure 6.10: ***Left to right, Top to bottom****: Example of Fusion with the proposed extensions failing to handle the splitting of a neural process across consecutive sections (the neural process in purple splits in two pieces, but Fusion does not link them correctly). One possible reason behind this failure is that the optical flow estimate between sections is inaccurate.*

the numerical results of the evaluation with the Rand Error and the Variation of Information of all the experiments in Figs. 6.11 and 6.12.

6.4 Discussion

In this chapter we have presented several extensions for Segmentation Fusion, the method we introduced in Chapter 5. Two of the proposed extensions, Bipartite Fusion and Nested Fusion allow Fusion to handle relatively large datasets, an important problem of interest in Connectomics, with divide-and-conquer schemes that are amenable for parallelization. We have also introduced an extension that allows Fusion to handle splits or branches of neural

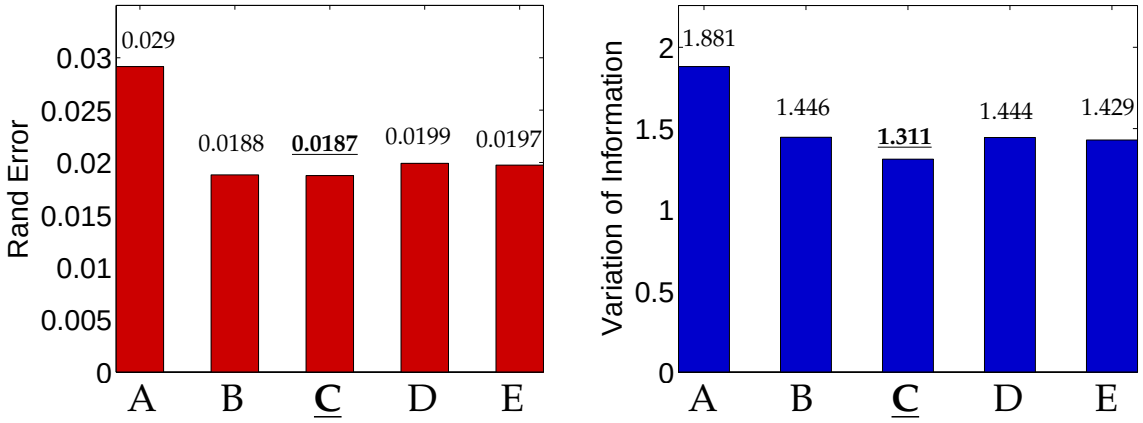


Figure 6.11: *Rand error and Variation of Information for our experiments on **5K**, A, B, C, D and E. Lower is better for both measures. Bipartite Fusion in 3D with branching (C) outperforms all other methods in both Rand Error and Variation of Information.*

processes in the segmentation.

In all our experiments, enabling Fusion to cope with branching (neural processes that split between sections) resulted in gains in reconstruction accuracy (see Experiments C, G and I). While the numerical improvement was timid, the difference was noticeable visually (e.g., as in Figures 6.7, 6.8 and 6.9).

We did not notice a substantial difference between Nested and Bipartite Fusion in our experiments. We speculate two possible reasons for this. One reason could be that the calibration of the parameters θ_s for the 3D segments with respect to the parameters θ_l in the cost function of Eq. 5.4 for Nested Fusion may be somewhat off. As opposed to with Fusion, in Nested Fusion some 3D segments (variables s_i in Eq. 5.4) have no links (variables l_j in Eq. 5.4) hiting them. For example, small 3D segments near the center of a subvolume may not have pixels

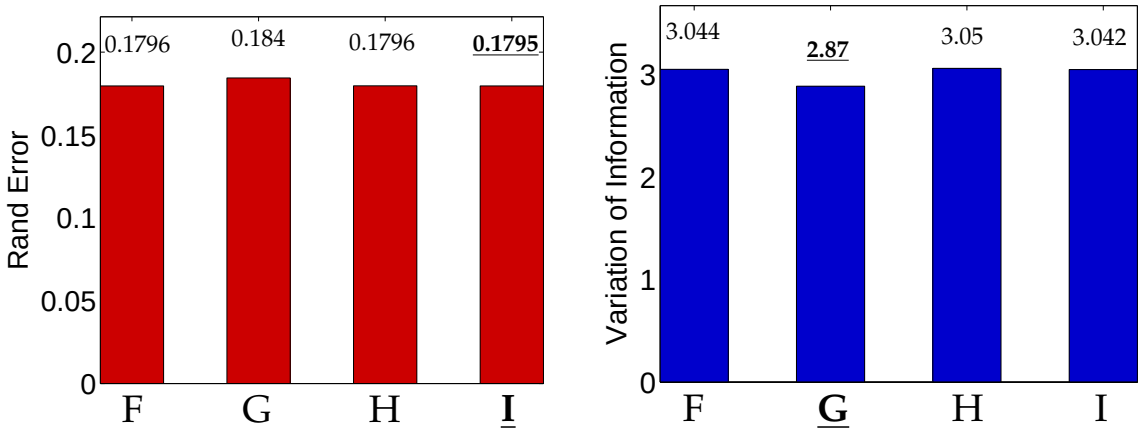


Figure 6.12: *Rand error and Variation of Information for our three experiments on **AC3**, G,H and I. Lower is better for both measures. Nested Fusion with branching (I) outperforms all other methods on the Rand Error, while Bipartite Fusion with branching (G) outperforms all other methods on the Variation of Information. As a reference, we obtained the lowest Rand Error (0.008) and lowest Variation of Information (0.87) when running Nested Fusion with branching using the 2D segments from the complete ground truth annotations available for **AC3**.*

on the boundaries of the subvolume, and therefore their selection would not be dependant of the selection of links between cubes. As we note in Section 7.1, the problem of calibrating the parameters θ_s and θ_l could be approached by learning these parameters from ground truth data. Another reason for the lack of improvement with Nested Fusion with respect to Bipartite Fusion is that we noticed CPLEX does not output substantially different 3D pre-segmentations in each subvolume (i.e., there is high redundancy between the multiple 3D segmentations for each subvolume). We believe this gives Nested Fusion little

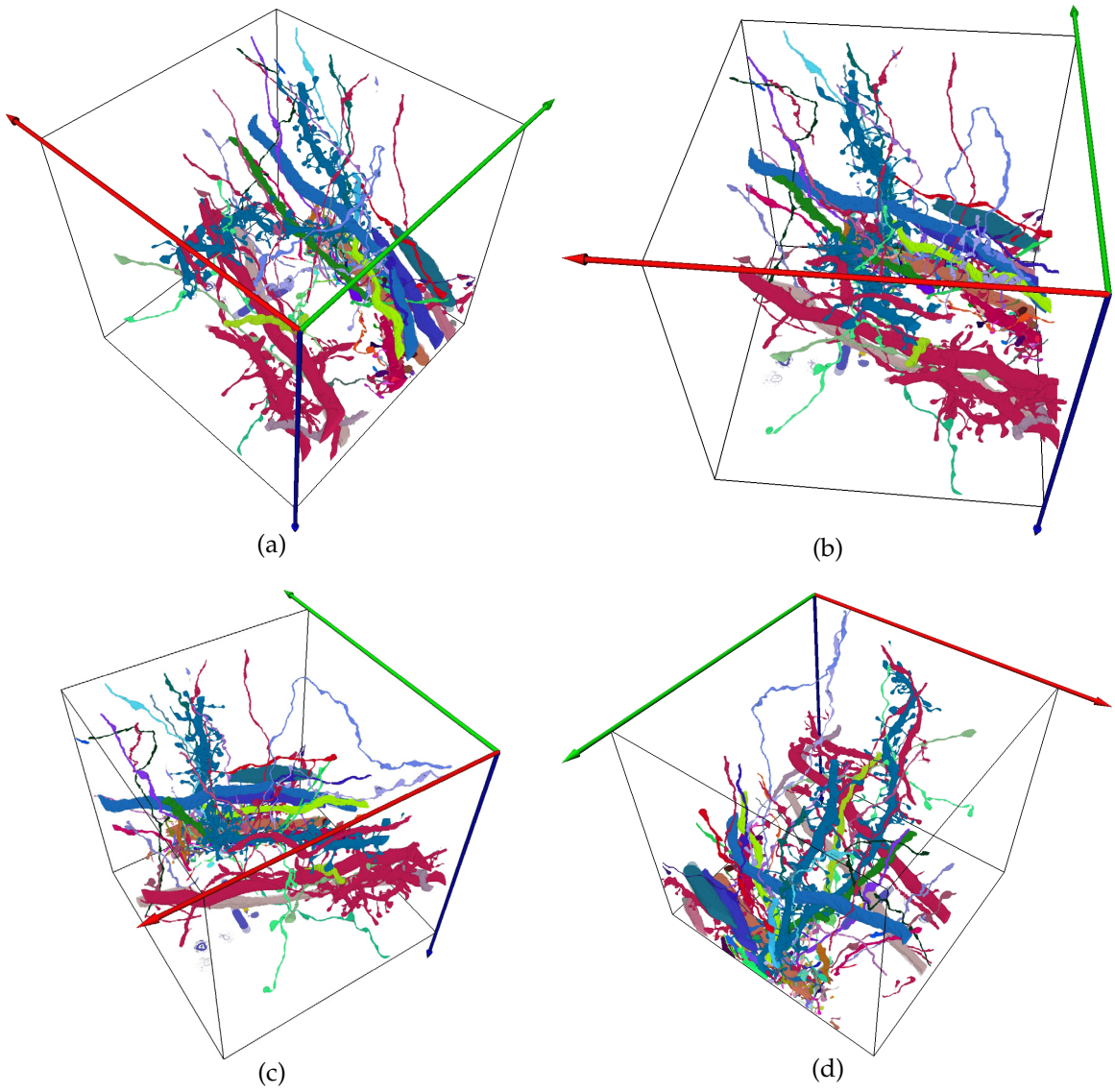


Figure 6.13: *Rendering of the ground truth on the dataset **5K**, a volume with 5,000 x 5,000 x 1000 voxels on which we evaluate our large scale reconstructions.*

room for improvement with respect to Bipartite Fusion.

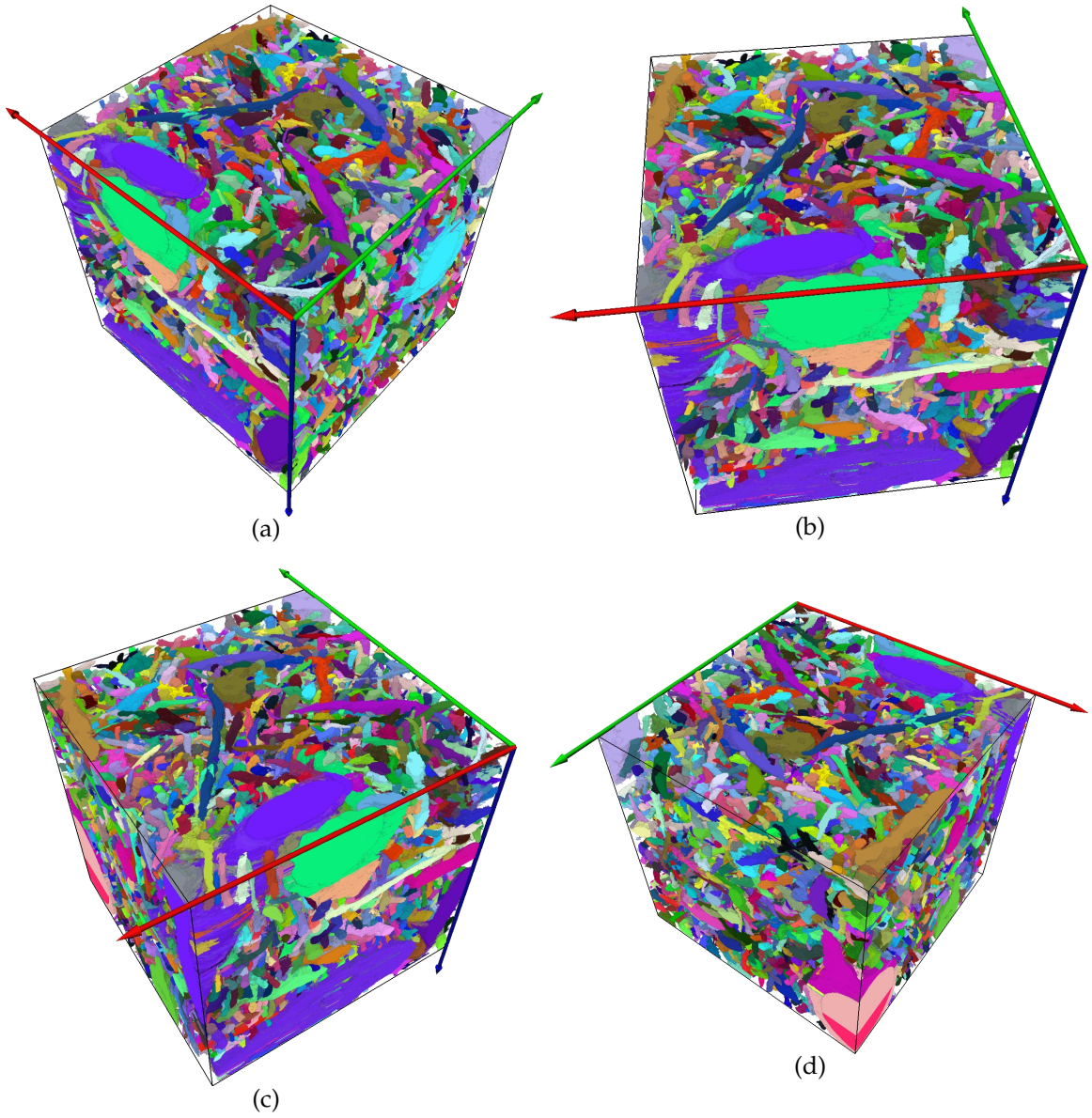
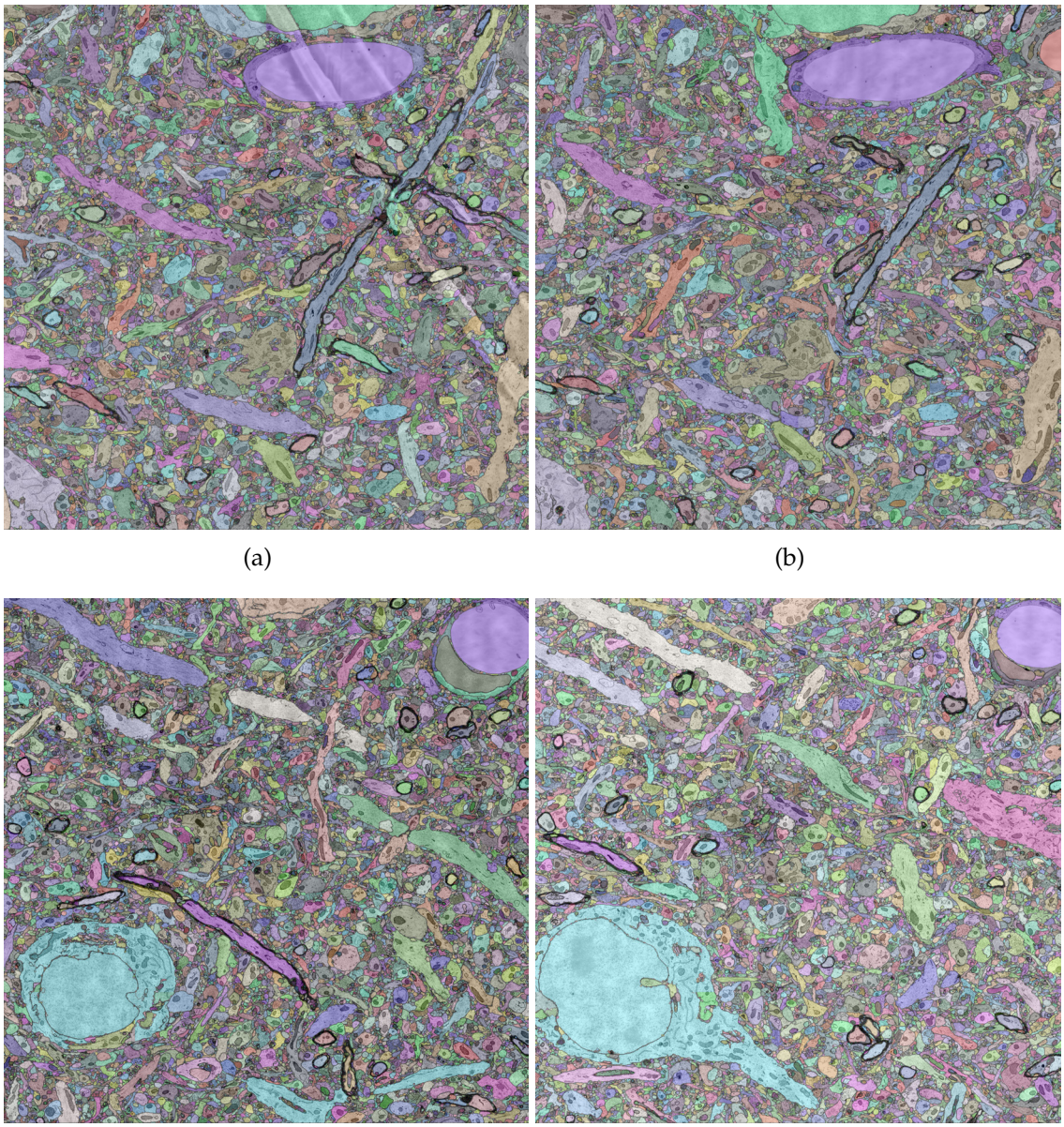


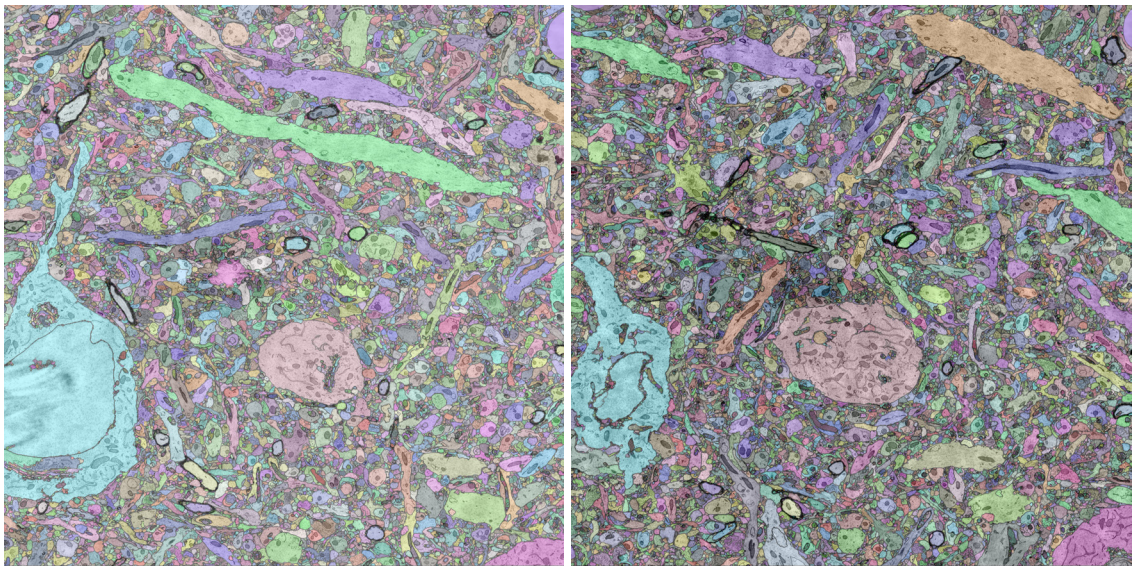
Figure 6.14: *Different 3D views of the automatic reconstruction results obtained in Experiment C. The dataset used for this experiment,* ***5K****, is a volume with 5,000 x 5,000 x 1000 voxels.*



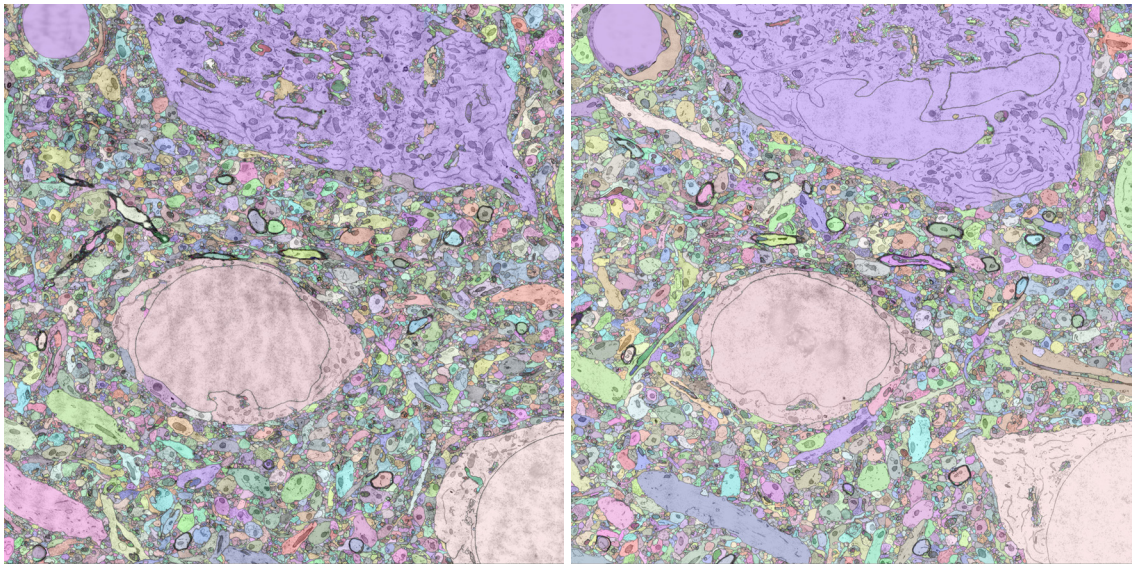
(a) (b)

(c) (d)

Figure 6.15: *Results of the segmentation from Experiment C. The dataset for this experiment,* ***5K****, is a volume with 5,000 x 5,000 x 1000 voxels. The sections displayed from left to right and top to bottom are 1, 20, 250 and 300.*



(a) (b)



(c) (d)

Figure 6.16: *Results of the segmentation from Experiment C. The dataset for this experiment, **5K**, is a volume with 5,000 x 5,000 x 1000 voxels. The sections displayed from left to right and top to bottom are 500, 551, 800 and 851. The two large cells in the bottom of subfigures (c) and (d) were automatically segmented with similar but different shades of brown (labels).*

Parallelization

As we mentioned earlier, the divide-and-conquer schemes we discussed in this chapter can be parallelized on a cluster or supercomputer according to the dependencies of the tasks in the processing pipelines of Figures 6.1 and 6.3. As a reference, Table 6.1 shows the number of jobs required to obtain the winning segmentation of Experiment C on the **5K** volume on the Harvard Odyssey Supercomputer[1] which is managed by the Job Scheduler IBM Platform LSF [2].

In all our experiments, the most significant bottleneck during processing was the communication of each of the workers with the filesystem (i.e., reading and writing data to disk). In Odyssey, there are two types of filesystems available, one based on NFS (Network File System) and one based in LUSTRE, which is known to deliver higher I/O performance than NFS[3] but is more expensive. When using NFS, the I/O bandwith limit of our processing was approximately 40 MB/s. In practice, this limited the number of workers we could have reading or writing simultaneously to disk in the cluster to only 20. The LUSTRE filesystem uses metadata servers that handle the data requests separate from the actual data transfers which are allowed to happen between multiple storage nodes and the computing node. In practice this allows LUSTRE to provide better I/O performance. In our experiments, we achieved an aggregated I/O bandwith

[1]Odyssey is managed by FAS Research Computing. For more details about its design and architecture see http://rc.fas.harvard.edu/

[2]IBM Platform LSF Job Scheduler: http://www.platform.com/workload-management/high-performance-computing/

[3]For details about their difference in performance see: http://wiki.lustre.org/index.php/NFS_vs._Lustre

Processing task	Number of jobs
2D pre-segmentations	16000
Fusion per subvolume	1136
Linear assignment problem	2824
Connected component analysis	1
Cube re-coloring	1136
Section stitching	1000

Table 6.1: *Number of parallel jobs required to obtain the segmentation result on a stack with 5,000 x 5,000 x 1000 pixels in Experiments B and C. The total number of jobs was 22097. Most of them run in less than 45 minutes. In total all jobs finished in approximately 2.5 days.*

of 500 MB when using LUSTRE. We believe that further improvements in computing scalability can be obtained with job scheduling schemes that maximize data locality, i.e., the use of job schedulers that try to allocate jobs to workers that are close to the data, while still balancing the overall I/O and CPU usage in the cluster. This functionality is not currently supported by Odyssey.

CHAPTER 7

Future Work and Conclusions

In this final chapter we outline promising avenues for further research and summarize the contributions of the thesis.

7.1 Future Work

We propose what we think are three promising extensions that can improve the quality of automatic segmentation as well as the effective time it takes to obtain circuit reconstructions.

"Smart Fusion" Methods

In the original Fusion formulation, each section of an EM stack is first pre-segmented in 2D. This is done multiple times to obtain multiple 2D pre-segmentations per section. The idea is to obtain a "bag" of possible 2D segments from which Fusion can later pick segments to connect in 3D. As we explained in Chapter 5, the 2D segmentations are obtained from the detection of cell membranes at the pixel level, and all possible segment-to-segment

comparisons are considered for connecting them in 3D, as long as they overlap in 2D across sections.

An important observation is that some of these segments may correspond to cross-sections of neural processes, while others may correspond to cross-sections of blood vessels or mitochondria. Some of them may also just correspond to regions of pixels between cells. In the original Fusion formulation there is no distinction between the source of the segments, and all possible segment-to-segment combinations are considered as feasible for the final 3D segmentation.

We believe one could significantly improve the quality of the reconstructions obtained with Fusion by training classifiers that examine segments and links of segments, and that discard those that one would not expect to see forming part of the reconstructed neural circuit. The prediction score or probability output of the classifiers can also replace the heuristics described in Chapter 5 for weighting the variables in the original optimization problem.

Finally, we also recommend exploring the connections between Fusion with clustering. For example, the recent work of [168] shows that "segmentation proposals", similar in spirit to the pre-segmentations we discussed for Segmentation Fusion, can be used to define a higher-order clustering problem for segmentation. In this formulation segments are encouraged or discouraged to group into clusters that align with the set of proposals, depending on the weight (positive or negative) assigned to the proposals.

Active Reconstruction

While methods like the ones proposed in this thesis can enable the automatic or semi-automatic 3D reconstruction of neural circuits, a non-trivial amount of human labor is still required to verify and proof-read (i.e., correct) the result of the segmentation. At the moment, this proof-reading and validation is necessary before neuroscientists can address questions about structure and connectivity in the reconstruction. The proof-reading can be highly time-consuming as neuroscientists or trained individuals need to visually inspect large portions of the stack to detect and correct relevant errors. Moreover, the most relevant structure, such as dendritic spines, is often sparsely distributed through the stack, making it difficult for the proof-reader to spot the regions in the stack that require most of her attention.

As the datasets get larger, we believe that the manual and blind inspection and proof-reading of the segmentation could slow down the process of circuit reconstruction, even as the accuracy of automatic segmentation procedures improve over time. To address this problem, we suggest the study of methods for neuron reconstruction and proof-reading based on the notion of *active inference*. Rather than requiring the user to blindly proof-read or blindly seed the reconstruction, we envision methods that ask the user for help on regions where the expected information about the reconstruction gained from the user's help is highest. These could be regions where the segmentation is most uncertain (e.g., because the image data is perhaps more noisy), and/or where the impact of making the wrong automatic decision, in terms of changes in the pixel

connectivity graph of the image stack (i.e., the number of pixels expected to change their label or color based on the automatic decision), is highest.

One way to implement such system would be to iteratively estimate the posterior distribution of the segmentation by asking questions to the user. Rather than fixing the posterior distribution and then estimating the optimal segmentation as the mode of the initial estimate of the posterior (as we did in Chapters 4 and 5), we envision a system where the computer starts with an estimate of the posterior distribution, and then builds a tree of user-friendly questions that divide the posterior in regions with equal probability mass at each split (i.e., questions that the computer is least confident about answering automatically). These questions could be restricted to relatively simple yes and no questions, such as asking if a set of edges should be present on the image (the user could answer the question by crossing those edges she finds to be wrong). The construction of the tree would be similar in spirit to clustering decision trees [169, 170]. The user would navigate this tree, iteratively answering questions, and the system would update the posterior accordingly, using a scheme similar to sequential Bayesian learning [81]. This process could proceed either until the user decides to stop, at which point we can rely on mode estimation for obtaining the most likely segmentation, or until the user reaches a leaf in the posterior (i.e., the posterior becomes a delta). Active inference methods like the one we just described would make the reconstruction more robust against errors in the initial estimates of the posterior, as well as to potential approximation errors in the optimization for estimating the mode of the

posterior.

We note that the problem of active clustering or active inference in general is closely related to the problem of active learning. Rather than constructing training samples for acquiring more ground truth data from the user, as it is customary in active learning, active inference methods can be seen as methods that query the user for the state of the unknown variables that are most informative in estimating the maximum a posteriori configuration in a statistical model [171, 172].

Distributed Segmentation Strategies

In the previous chapter, we discussed two schemes Bipartite Fusion and Nested Fusion for scaling Fusion by dividing a large image stack into smaller volumes that can be segmented independently by independent workers (e.g., nodes in a cluster). The independent segmentations were then combined together into a final segmentation for the full volume. In these schemes, the subvolumes had approximately the same fixed size, and this size was determined *a priori* regardless of differences in local image quality or the local complexity of underlying neural circuit. The advantage of this approach is its simplicity, one can subdivide the original segmentation problem into smaller parallelizable subproblems by dividing the images into equal-sized subvolumes that are loaded by each worker.

However, in our experiments we noticed that dividing a large volume by fixed-size subvolumes resulted in having some workers finish segmenting

subvolumes in much less time than others, resulting in the underutilization of computing resources. Indeed, the complexity of the data can vary greatly through the stack, for example regions with vessels are easier to segment by Fusion than regions with a dense set of thin neural processes, where more segments and links (i.e., variables to optimize for) may be needed. If one aims to maximize reconstruction throughput, a more reasonable divide-and-conquer approach would be distribute the workload evenly through the workers. Ideally one we divide volumes according to expected time required to segment it, or perhaps the number of variables one aims to solve in each subproblem. We think that the study of methods that divide the segmentation of large sequences according to the local complexity of the segmentation is a promising direction of research.

We believe that a promising direction of research to address these questions for dividing may be the study of decomposition methods for large scale optimization problems such as Dual Decomposition and The Alternating Direction Method of Multipliers [173–175].

7.2 Conclusions

In this thesis we have addressed the problem of segmenting electron microscopy stacks for neuron reconstruction. We have proposed solutions that are suitable for interactive and automatic reconstruction, and explored the connection between neuron reconstruction with video segmentation.

In Chapter 3 we looked at the applicability of an important class of image segmentation methods, level set methods, for segmenting neural processes. We showed that conventional level set methods cannot reliably capture the cellular membranes of individual processes in a single partition. Motivated by the fact that neuronal cross-sections appear as collections of closed contours with dark membranes on electron micrographs, we proposed a novel framework to control the geometric layout of level set partitions. We then used this framework to propose *Active Ribbons*, a deformable geometric model for segmenting the cellular membranes of individual neurons. As far as we know, this is the first deformable model designed to specifically capture ribbon-looking objects in images, and the first semi-automatic method for 3D neuron reconstruction from EM images.

The work presented in Chapter 3 was published in [46] and served as the foundation of NeuroTrace [45, 124], a tool developed in our lab that extended Active Ribbons to work on GPU hardware and in 3D. In [124] we conducted a user study that involved two novice users and four neuroscientists, and observed that users systematically reconstructed neural processes with NeuroTrace in half the time it took them to do it with Reconstruct [37].

Motivated by the connection between connectomics and video segmentation described in Section 1.3, in Chapter 4 we proposed what we think is the first solution for the problem of unsupervised multi-frame *on-line* video segmentation. In order segment a video with an unknown number of objects (labels), in MHVS we first enumerate possible time trajectories of 2D regions of

pixels that serve as candidate pixel labels. We then let these trajectories compete with each other on a higher-order random field to label individual pixels within a sliding window of frames. This allows MHVS to segment arbitrarily long videos while encouraging label consistency between more than two frames.

In an effort to propose a solution for 3D segmentation that avoids the explicit discovery of labels of MHVS, in Chapter 5 we introduced Segmentation Fusion for 3D neuron reconstruction from anisotropic EM stacks. The main idea in Fusion is to formulate a binary optimization problem that can identify globally optimal combinations of candidate 2D cross-sections of neural processes across multiple sections. To obtain the 2D candidates, we introduced a novel set of features for detecting cell membranes with a pixel classifier that take advantage of the rotation invariance in ssSEM stacks. In contrast to other descriptors based on the careful manual selection of filter banks, we showed that one can obtain state-of-the-art pixel classification results with a general purpose descriptor based on disk harmonics. Our features also allow for the reconstruction of patches directly from the descriptor, which has the benefit of facilitating the visual inspection of the information that the pixel classifier "sees".

Finally, we discussed several schemes for scaling and improving Fusion, and outlined some directions for future work. We sketched ideas that can help with the weighting and selection of segments and links for Fusion, and proposed the use of Dual Decomposition and The Alternating Direction Method of Multipliers to parallelize the reconstruction of large neural circuits. We also outlined an active inference method that we think could significantly reduce the burden of

manually proofreading and validating large reconstructions.

Overall, we believe this thesis advances the state-of-the-art in automating neuron reconstruction from anisotropic electron microscopy stacks, and prepares the ground for the comparative analysis of large neural circuits. We have proposed novel methods for semi-automatic neuron reconstruction, scalable automatic neuron reconstruction, and the multi-frame on-line segmentation of videos. On a more technical side, we have also proposed new techniques for inducing specific geometric layouts of partitions with level set functions, and a novel statistical graphical model that can fuse information from multiple 2D segmentation sources for 3D segmentation.

We believe these methods represent a substantial contribution to the connectomics and related computer vision literature, and are optimistic that continued efforts along the proposed lines of future work will enable the faster reconstruction of increasingly larger neural circuits.

APPENDIX A

A.1 Linear Programming Formulation for the Ising Model

We provide an example of how to formulate a linear program for a MAP-MRF problem. We consider the MAP problem of the Ising model we described in Eq. 2.21:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} \left\{ \sum_{i \in \mathcal{V}} \theta_K \cdot \mathbb{I}(x_i = x_j) + \sum_{i \in \mathcal{E}} \left(\theta_{i_0} \cdot \mathbb{I}(x_i = 0) + \theta_{i_1} \cdot \mathbb{I}(x_i = 1) \right) \right\} \quad \text{(A.1)}$$

This problem was formulated as a quadratic pseudo-boolean optimization (QPBO) problem in Eq. 2.22, but as we show next, it can also be formulated as a linear binary programming problem. Note that we can rewrite Eq. A.1 as:

$$\begin{aligned} \mathbf{x}^* = \arg\max_{\mathbf{x}} \Bigg\{ & \sum_{i \in \mathcal{V}} (\theta_K \cdot \mathbb{I}_{i,j}(1,1) + \theta_K \cdot \mathbb{I}_{i,j}(0,0)) + \\ & \sum_{i \in \mathcal{E}} \theta_{i_0} \cdot \mathbb{I}(x_i = 0) + \theta_{i_1} \cdot \mathbb{I}(x_i = 1) \Bigg\}, \end{aligned} \quad \text{(A.2)}$$

where the function $\mathbb{I}_{i,j}(a,b)$ is defined as $\mathbb{I}_{i,j}(a,b) = \mathbb{I}\left(x_i = a \;\&\; x_j = b\right)$, i.e., it returns 1, if the condition $x_i = a$ & $x_j = b$ is true, and 0 otherwise. Note that

$\mathbb{I}_{i,j}(a,b)$ must satisfy:

$$\begin{aligned}\sum_b \mathbb{I}_{i,j}(1,b) &= \mathbb{I}(x_i = 1)\\ \sum_a \mathbb{I}_{i,j}(a,1) &= \mathbb{I}(x_j = 1)\\ \sum_{a,b} \mathbb{I}_{i,j}(a,b) &= 1.\end{aligned} \tag{A.3}$$

To see the above, note for example that $x_i = 1$ is true if and only if either the condition $x_i = 1$ & $x_j = 1$ is true or the condition $x_i = 1$ & $x_j = 0$ is true, which is captured by the first of the equations above.

Since the conditions from Eq. A.3 must hold, we can add them as constraints to the problem in Eq. A.2 and treat the indicator functions $\mathbb{I}_{i,j}(a,b)$ as additional (auxiliary) variables in the optimization. Since we have that $\mathbb{I}(x_i = 1) = x_i$ and $\mathbb{I}(x_i = 0) = 1 - x_i$, we can rewrite A.1 to have a cost function that is linear in the variables x_i's that are to be inferred, resulting in the following binary linear programming problem:

$$\begin{aligned}\mathbf{x}^* = \arg\max_{\mathbf{x}} \Bigg\{ &\sum_{i\in\mathcal{V}} (\theta_K \cdot \mathbb{I}_{i,j}(1,1) + \theta_K \cdot \mathbb{I}_{i,j}(0,0)) + \\ &\sum_{i\in\mathcal{E}} \theta_{i_0} \cdot (1 - x_i) + \theta_{i_1} \cdot x_i \Bigg\}\\ \text{such that:}\quad &\sum_b \mathbb{I}_{i,j}(1,b) = x_i\\ &\sum_a \mathbb{I}_{i,j}(a,1) = x_j\\ &\sum_{a,b} \mathbb{I}_{i,j}(a,b) = 1.\end{aligned} \tag{A.4}$$

Note that as before, we are still optimizing over $\mathbf{x}$, but we have transformed the indicator functions into auxiliary variables that depend on the value of $\mathbf{x}$ through

new linear constraints that we added to the problem. In other words, we have made the optimization problem (the cost function) linear by increasing the dimensionality of the original problem.

The derivation above shows that the MAP inference of a binary pairwise MRF model such as the Ising model can be formulated as a linear programming problem. Similar transformations can be used to formulate MAP inference on general MRFs as linear programs. We refer the reader to [176] for a recent study that discusses general methods for formulating LP problems from QPBO formulations, which are common in the MAP-MRF literature. We also note that in some cases, such as in the segmentation method we proposed in Chapter 5, such transformations are not necessary since the problem of MAP inference can be directly expressed as a linear program.

Bibliography

[1] N. Kasthuri and J. W. Lichtman, "Neurocartography," *Neuropsychopharmacology*, vol. 35, no. 1, pp. 342–343, 2010.

[2] G. S. Wig, B. L. Schlaggar, and S. E. Petersen, "Concepts and principles in the analysis of brain networks," *Annals of the New York Academy of Sciences*, vol. 1224, no. 1, pp. 126–146, 2011.

[3] J. W. Lichtman and J. R. Sanes, "Ome sweet ome: what can the genome tell us about the connectome?" *Current Opinion in Neurobiology*, vol. 18, no. 3, pp. 346–353, 2008.

[4] S. Seung, *Connectome: How the Brain's Wiring Makes Us Who We Are*. Houghton Mifflin Harcourt, 2012.

[5] H. S. Seung, "Reading the book of memory: Sparse sampling versus dense mapping of connectomes," *Neuron*, vol. 62, no. 1, pp. 17–29, 2009.

[6] J. G. White, E. Southgate, J. N. Thomson, and S. Brenner, "The structure of the nervous system of the nematode caenorhabditis elegans," *Philosophical*

Transactions of the Royal Society B Biological Sciences, vol. 314, no. 1165, pp. 1–340, 1986.

[7] D. D. Bock, W. A. Lee, A. M. Kerlin, M. L. Andermann, G. Hood, A. W. Wetzel, S. Yurgenson, E. R. Soucy, H. S. Kim, and R. C. Reid, "Network anatomy and in vivo physiology of visual cortical neurons," *Nature*, vol. 471, no. 7337, pp. 177–182, Mar. 2011.

[8] K. Hayworth, N. Kasthuri, R. Schalek, and J. Lichtman, "Automating the collection of ultrathin serial sections for large volume TEM reconstructions," *Microscopy and Microanalysis*, vol. 12, no. Supplement S02, pp. 86–87, 2006.

[9] D. B. Chklovskii, S. Vitaladevuni, and L. K. Scheffer, "Semi-automated reconstruction of neural circuits using electron microscopy," *Current Opinion in Neurobiology*, vol. 20, no. 5, pp. 667 – 675, 2010.

[10] S. Rieger and R. W. Köster, "Preparation of zebrafish embryos for transmission electron microscopy," *Cold Spring Harbor Protocols*, vol. 2007, no. 6, Jun. 2007.

[11] T. Hey, S. Tansley, and K. Tolle, Eds., *The Fourth Paradigm: Data-Intensive Scientific Discovery: Discovering the Wiring Diagram of the Brain*. Microsoft Research, 2009.

[12] M. Helmstaedter, K. L. Briggman, and W. Denk, "3d structural imaging of the brain with photons and electrons." *Current Opinion in Neurobiology*, vol. 18, no. 6, pp. 633–641, 2008.

[13] V. Jain, H. S. Seung, and S. C. Turaga, "Machines that learn to segment images: a crucial technology for connectomics." *Current Opinion in Neurobiology*, vol. 20, no. 5, pp. 653–666, 2010.

[14] A. Vazquez-Reina, M. Gelbart, D. Huang, J. Lichtman, E. Miller, and H. Pfister, "Segmentation fusion for connectomics," in *Proceedings of the 13th International Conference on Computer Vision (ICCV 2011)*.

[15] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, Sep. 2004.

[16] D. A. Tolliver and G. L. Miller, "Graph partitioning by spectral rounding: Applications in image segmentation and clustering," in *Proceedings of the 19th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, 2006.

[17] E. Sharon, M. Galun, D. Sharon, R. Basri, and A. Brandt, "Hierarchy and adaptivity in segmenting visual scenes," *Nature*, vol. 442, no. 7104, p. 810, Jun. 2006.

[18] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[19] D. Glasner, S. Vitaladevuni, and R. Basri, "Contour-based joint clustering of multiple segmentations."

[20] P. Arbelaez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik, "Semantic segmentation using regions and parts," in *Proceedings of the 25th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012)*.

[21] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898 –916, 2011.

[22] P. Dollar, Z. Tu, and S. Belongie, "Supervised learning of edges and object boundaries," in *Proceedings of the 19th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2006)*.

[23] M. Helmstaedter and P. P. Mitra, "Computational methods and challenges for large-scale circuit mapping," *Current Opinion in Neurobiology*, vol. 22, no. 1, pp. 162 – 169, 2012.

[24] Y. Kubota, S. Hatada, and Y. Kawaguchi, "Important factors for the three-dimensional reconstruction of neuronal structures from serial ultrathin sections," *Frontiers in Neural Circuits*, vol. 3, no. 4, p. 12, 2009.

[25] V. Jain, "Machine learning of image analysis with convolutional networks and topological constraints," Ph.D. dissertation, Massachusetts Institute of Technology, USA, 2009.

[26] V. Kaynig, T. J. Fuchs, and J. M. Buhmann, "Geometrical consistent 3D tracing of neuronal processes in sstem data," in *Proceedings of the 13th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI 2010)*.

[27] V. Jain, B. Bollmann, M. Richardson, D. Berger, M. Helmstaedter, K. Briggman, W. Denk, J. Bowden, J. Mendenhall, W. Abraham, K. Harris, N. Kasthuri, K. Hayworth, R. Schalek, J. Tapia, J. Lichtman, and H. Seung, "Boundary learning by optimization with topological constraints," in *Proceedings of the 23rd IEEE Conference in Computer Vision and Pattern Recognition (CVPR 2010)*.

[28] R. Kumar, A. Vázquez-Reina, and H. Pfister, "Radon-like features and their application to connectomics," in *Proceedings of the 23rd IEEE Conference in Computer Vision and Pattern Recognition Workshops (CVPRW 2010)*.

[29] A. Lucchi, K. Smith, R. Achanta, V. Lepetit, and P. Fua, "A fully automated approach to segmentation of irregularly shaped cellular structures in EM images," in *Proceedings of the 13th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI 2010)*.

[30] V. Kaynig, T. Fuchs, and J. Buhmann, "Neuron geometry extraction by perceptual grouping in ssTEM images," in *Proceedings of the 23rd*

International Conference in Computer Vision and Pattern Recognition (CVPR 2010).

[31] S. Vitaladevuni and R. Basri, "Co-clustering of image segments using convex optimization applied to em neuronal reconstruction," in *Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*.

[32] Y. Mishchenko, "Automation of 3D reconstruction of neural tissue from large volume of conventional serial section transmission electron micrographs," *Journal of Neuroscience Methods*, vol. 176, no. 2, pp. 276–289, Jan. 2009.

[33] E. Jurrus, M. Hardy, T. Tasdizen, P. Fletcher, P. Koshevoy, C. Chien, W. Denk, and R. Whitaker, "Axon tracking in serial block-face scanning electron microscopy," *Medical Image Analysis*, vol. 13, no. 1, pp. 180–188, Feb. 2009.

[34] M. Roberts, W.-K. Jeong, A. Vázquez-Reina, M. Unger, H. Bischof, J. Lichtman, and H. Pfister, "Neural process reconstruction from sparse user scribbles," in *Medical Image Computing and Computer Assisted Intervention (MICCAI '11)*, 2011, pp. 621–628.

[35] Y. Pan, W.-K. Jeong, and R. T. Whitaker, "Markov surfaces: A probabilistic framework for user-assisted three-dimensional image segmentation." *Computer Vision and Image Understanding*, vol. 115, no. 10, pp. 1375–1383, 2011.

[36] M. Unger, T. Mauthner, T. Pock, and H. Bischof, "Tracking as segmentation of spatial-temporal volumes by anisotropic weighted TV," in *Proceedings of the 7th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR '09)*. Berlin, Heidelberg: Springer-Verlag, pp. 193–206.

[37] J. C. Fiala, "Reconstruct: a free editor for serial section microscopy," *Journal of microscopy*, vol. 218, no. Pt 1, pp. 52–61, Apr. 2005.

[38] Y. Mishchenko, T. Hu, J. Spacek, J. Mendenhall, K. M. Harris, and D. B. Chklovskii.

[39] G. W. Knott, A. Holtmaat, L. Wilbrecht, E. Welker, and K. Svoboda, "Spine growth precedes synapse formation in the adult neocortex in vivo," *Nature Neuroscience*, vol. 9, no. 9, p. 1117, Aug. 2006.

[40] A. Cardona, "TrakEM2: an ImageJ-based program for mophological data mining and 3D modeling," in *Proceedings of the ImageJ User and Developer Conference*, 2006, pp. 43–51.

[41] A. Cardona, S. Saalfeld, S. Preibisch, B. Schmid, A. Cheng, J. Pulokas, P. Tomancak, and V. Hartenstein, "An integrated micro- and macroarchitectural analysis of the drosophila brain by computer-assisted serial section electron microscopy," *PLoS Biol*, vol. 8, no. 10, p. e1000502, 10 2010.

[42] C. Sommer, C. N. Straehle, U. Köthe, and F. A. Hamprecht, "Ilastik: Interactive learning and segmentation toolkit," in *Proceedings of the 8th International Symposium on Biomedical Imaging (ISBI 2011)*, pp. 230–233.

[43] M. Helmstaedter, K. L. Briggman, and W. Denk, "High-accuracy neurite reconstruction for high-throughput neuroanatomy," *Nature Neuroscience*, vol. 14, no. 8, pp. 1081–1088, 2011.

[44] K. L. Briggman, M. Helmstaedter, and W. Denk, "Wiring specificity in the direction-selectivity circuit of the retina," *Nature*, vol. 471, no. 7337, pp. 183–188, mar 2011.

[45] W.-K. Jeong, J. Beyer, M. Hadwiger, R. Blue, C. Law, A. Vázquez-Reina, C. Reid, J. Lichtman, and H. Pfister, "SSECRETT and NeuroTrace: Interactive visualization and analysis tools for large-scale neuroscience datasets," *IEEE Computer Graphics and Applications*, vol. 30, pp. 58–70, 2010.

[46] A. Vázquez-Reina, E. Miller, and H. Pfister, "Multiphase geometric couplings for the segmentation of neural processes," in *Proceedings of the 22nd International Conference on Computer Vision and Pattern Recognition (CVPR 2009)*.

[47] M. J. Wainwright and M. I. Jordan, *Graphical Models, Exponential Families, and Variational Inference*. Hanover, MA, USA: Now Publishers Inc., 2008.

[48] J.-M. Morel and S. Solimini, *Variational Methods in Image Segmentation*. Birkhauser, 1995.

[49] D. Cremers, "Statistical shape knowledge in variational image segmentation," Ph.D. dissertation, Department of Mathematics and Computer Science, University of Mannheim, Germany, 2002.

[50] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient n-Dp image segmentation," *International Journal of Computer Vision*, vol. 70, no. 2, pp. 109–131, Nov. 2006.

[51] A. Tsai, J. Yezzi, A., W. Wells, C. Tempany, D. Tucker, A. Fan, W. Grimson, and A. Willsky, "A shape-based approach to the segmentation of medical imagery using level sets," *IEEE Transactions on Medical Imaging*, vol. 22, no. 2, pp. 137 –154, feb. 2003.

[52] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, Feb. 1997.

[53] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *International Journal of Computer Vision*, vol. 81, no. 1, pp. 2–23, 2009.

[54] J. Malcolm, Y. Rathi, and A. Tannenbaum, "Graph cut segmentation with nonlinear shape priors," in *Proceedings of the 14th international conference on Image processing (ICIP 2007)*.

[55] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[56] G. Sapiro, *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, 2006.

[57] S. Osher and N. Paragios, *Geometric Level Set Methods in Imaging, Vision, and Graphics*. Springer, 2003.

[58] R. Kimmel, *Numerical Geometry of Images: Theory, Algorithms, and Applications*. SpringerVerlag, 2003.

[59] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on Pure and Applied Mathematics*, vol. 42, no. 5, pp. 577–685, 1989.

[60] L. A. Vese and T. F. Chan, "A multiphase level set framework for image segmentation using the mumford and shah model," *International Journal of Computer Vision*, vol. 50, no. 3, pp. 271–293, 2002.

[61] T. F. Chan, M. Nikolova, and S. Esedoglu, "Algorithms for finding global minimizers of image segmentation and denoising models," *SIAM Journal on Applied Mathematics*, vol. 66, no. 5, pp. 1632–1648, 2006.

[62] T. E. Chan and L. A. Vese, "A level set algorithm for minimizing the mumford-shah functional in image processing," in *IEEE Workshop on Variational and Level Set Methods*, 2001, p. 161.

[63] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, Feb. 2001.

[64] T. F. Chan, J. Shen, and L. Vese, "Variational PDE models in image processing," 2002 Joint Mathematics Meeting.

[65] A. Vasilevskiy and K. Siddiqi, "Flux maximizing geometric flows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1565–1578, 2002.

[66] B. Li and S. T. Acton, "Active contour external force using vector field convolution for image segmentation," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2096–2106, 2007.

[67] C. Xu and J. L. Prince, "Generalized gradient vector flow external forces for active contours," *Signal Processing*, vol. 71, pp. 131–139, 1998.

[68] X. Xie and M. Mirmehdi, "MAC: Magnetostatic active contour model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 4, pp. 632–646, 2008.

[69] S. C. Zhu and A. Yuille, "Region competition: unifying snakes, region growing, and bayes/MDL for multiband image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, pp. 884 –900, sep 1996.

[70] C. Sagiv, N. A. Sochen, and Y. Y. Zeevi, "Integrated active contours for texture segmentation," *IEEE Transactions on Image Processing*, vol. 1, pp. 1–19, 2004.

[71] J. Kim, I. Fisher, J.W., A. Yezzi, M. Cetin, and A. Willsky, "A nonparametric statistical method for image segmentation using information theory and curve evolution," *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1486 –1502, oct. 2005.

[72] D. Cremers, M. Rousson, and R. Deriche, "A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape," *International Journal of Computer Vision*, vol. 72, no. 2, pp. 195–215, Apr. 2007.

[73] E. Sudderth, "Graphical models for visual object recognition and tracking," Ph.D. dissertation, Massachusetts Institute of Technology, USA, 2006.

[74] D. Sontag, "Approximate inference in graphical models using LP relaxations," Ph.D. dissertation, Massachusetts Institute of Technology, USA, 2010.

[75] S. Z. Li, *Markov Random Field Modeling in Image Analysis*, 3rd ed. Springer, 2009.

[76] A. Blake, P. Kohli, and C. Rother, *Markov Random Fields for Vision and Image Processing*. The MIT Press, 2011.

[77] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, p. 2001, 2001.

[78] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, Nov. 1984.

[79] O. Veksler, "Efficient graph-based energy minimization methods in computer vision," Ph.D. dissertation, Cornell University, USA, 1999.

[80] J. Besag, "On the statistical analysis of dirty pictures," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 48, no. 3, pp. pp. 259–302, 1986.

[81] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2007.

[82] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A comparative study of energy minimization methods for markov random fields with smoothness-based priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 1068 –1080, 2008.

[83] C. Yanover, T. Meltzer, Y. Weiss, P. Bennett, and E. Parrado-hernÃąndez, "Linear programming relaxations and belief propagation âĂŞ an empirical study," *Journal of Machine Learning Research*, vol. 7, p. 2006, 2006.

[84] H. Ishikawa, "Transformation of general binary MRF minimization to the first-order case," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 1234–1249, 2011.

[85] S. Ramalingam, p. Kohli, K. Alahari, and P. H. S. Torr, "Exact inference in multi-label CRFs with higher order cliques." in *Proceedings of the 21st IEEE Conference in Computer Vision and Pattern Recognition (CVPR 2008)*, 2008.

[86] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 65–81, 2004.

[87] D. M. Greig, B. T. Porteous, and A. H. Seheult, "Exact maximum a posteriori estimation for binary images," *Journal of the Royal Statistical Society Series B Methodological*, vol. 51, no. 2, pp. 271–279, 1989.

[88] D. Bertsekas, *Network Optimization Continuous and Discrete Models*. Athena Scientific, 1998.

[89] E. Borros, P. L. Hammer, and X. Sun, "Network flows and minimization of quadratic pseudo-boolean functions," *RUTCOR Research Report*, no. RRR 17-1991, 1991.

[90] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer, "Optimizing binary MRFs via extended roof duality," in *Proceedings of the 20th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007)*.

[91] V. Lempitsky, C. Rother, S. Roth, and A. Blake, "Fusion moves for markov random field optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1392–1405, aug 2010.

[92] D. Freedman and P. Drineas, "Energy minimization via graph cuts: Settling what is possible," in *Proceedings of the 18th Conference in Computer Vision and Pattern Recognition (CVPR 2005)*.

[93] P. Kohli, L. Ladický, and P. H. Torr, "Robust higher order potentials for enforcing label consistency," *International Journal of Computer Vision*, vol. 82, no. 3, pp. 302–324, 2009.

[94] C. Rother, P. Kohli, W. Feng, and J. Jia, "Minimizing sparse higher order energy functions of discrete variables." in *Proceedings of the 22nd Conference in Computer Visiona and Pattern Recognition (CVPR 2009)*.

[95] J. Yedidia, W. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Transactions on Information Theory*, vol. 51, no. 7, pp. 2282 – 2312, july 2005.

[96] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1568 –1583, oct. 2006.

[97] Y. Weiss, C. Yanover, and T. Meltzer, "Map estimation, linear programming and belief propagation with convex free energies," *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence (UAI 2007)*, pp. 416–425.

[98] M. Wainwright, T. Jaakkola, and A. Willsky, "MAP estimation via agreement on trees: message-passing and linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 11, pp. 3697 – 3717, nov. 2005.

[99] D. Bertsimas and J. Tsitsiklis, *Introduction to Linear Optimization*, 1st ed. Athena Scientific, 1997.

[100] D. Bertsimas and R. Weismantel, *Optimization over Integers*. Dynamic Ideas, 2005.

[101] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1988.

[102] N. Komodakis and G. Tziritas, "Approximate labeling via graph cuts based on linear programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1436 –1453, aug. 2007.

[103] J. Yezzi, A., A. Tsai, and A. Willsky, "A statistical approach to snakes for bimodal and trimodal imagery," in *Proceedings of the 7th International Conference on Computer Vision (ICCV 1999)*.

[104] T. Brox and J. Weickert, "Level set segmentation with multiple regions," *IEEE Transactions on Image Processing*, vol. 15, no. 10, pp. 3213–3218, 2006.

[105] X. Fan, P.-L. Bazin, and J. Prince, "A multi-compartment segmentation framework with homeomorphic level sets," in *Proceedings of the 21st IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008)*.

[106] K. Fundana, N. C. Overgaard, and A. Heyden, "Variational segmentation of image sequences using region-based active contours and deformable

shape priors," *International Journal of Computer Vision*, vol. 80, no. 3, pp. 289–299, 2008.

[107] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: a level set approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 2, pp. 158–175, 1995.

[108] M.Rousson and N.Paragios, "Prior knowledge, level set representations & visual grouping," *International Journal of Computer Vision*, vol. 76, no. 3, pp. 231–243, 2008.

[109] D. Cremers, S. J. Osher, and S. Soatto, "Kernel density estimation and intrinsic alignment for shape priors in level set segmentation," *International Journal of Computer Vision*, vol. 69, no. 3, pp. 335–351, 2006.

[110] J. Kim, M. Çetin, and A. S. Willsky, "Nonparametric shape priors for active contour-based image segmentation," *Signal Processing*, vol. 87, no. 12, pp. 3021–3044, 2007.

[111] A. Tsai, W. Wells, C. Tempany, E. Grimson, and A. Willsky, "Mutual information in coupled multi-shape model for medical image segmentation." *Medical Image Analysis*, vol. 8, no. 4, pp. 429–445, 2004.

[112] A. Litvin and W. C. Karl, "Coupled shape distribution-based segmentation of multiple objects," in *Proceedings of the 19th international conference on Information Processing in Medical Imaging (IPMI 2005)*.

[113] N. Vu and B. Manjunath, "Shape prior segmentation of multiple objects with graph cuts," in *Proceedings of the 21st IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008)*.

[114] C. Samson, L. Blanc-Féraud, G. Aubert, and J. Zerubia, "Level set model for image classification," *International Journal of Computer Vision*, vol. 40, no. 3, pp. 187–197, 2000.

[115] N. Paragios and R. Deriche, "Geodesic active regions and level set methods for supervised texture segmentation," *International Journal of Computer Vision*, vol. 46, no. 3, pp. 223–247, 2002.

[116] ——, "Coupled geodesic active regions for image segmentation: A level set approach," in *Proceedings of the 6th European Conference on Computer Vision (ECCV 2000)*.

[117] H.-K. Zhao, T. Chan, B. Merriman, and S. Osher, "A variational level set approach to multiphase motion," *Journal of Computational Physics*, vol. 127, no. 1, pp. 179–195, 1996.

[118] A. Yezzi, A. Tsai, and A. Willsky, "A fully global approach to image segmentation via coupled curve evolution equations," *Journal of Visual Communication and Image Representation*, vol. 13, no. 1-2, pp. 195–216, 2002.

[119] D. Cremers, N. Sochen, and C. Schnörr, "A multiphase dynamic labeling model for variational recognition-driven image segmentation," *International Journal of Computer Vision*, vol. 66, no. 1, pp. 67–81, 2006.

[120] T. Chan and W. Zhu, "Level set based shape prior segmentation," in *Proceedings of the 18th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, 2005.

[121] M. Fussenegger, R. Deriche, and A. Pinz, "A multiphase level set based segmentation framework with pose invariant shape priors," in *Proceedings of the 7th Asian Conference on Computer Vision (ACCV 2006)*, pp. 395–404.

[122] P. W. Michor and D. Mumford, "Riemannian geometries on spaces of plane curves," *Journal of the European Mathematical Society*, vol. 8, pp. 1–48, 2006.

[123] R. Aharoni, A. Herschkovitz, R. Eilam, M. Blumberg-Hazan, M. Sela, W. Bruck, and R. Arnon, "Demyelination arrest and remyelination induced by glatiramer acetate treatment of experimental autoimmune encephalomyelitis," *Proceedings of the National Academy of Sciences*, vol. 105, no. 32, pp. 11 358–11 363, 2008.

[124] W.-K. Jeong, J. Beyer, M. Hadwiger, A. Vázquez-Reina, H. Pfister, and R. T. Whitaker, "Scalable and interactive segmentation and visualization of neural processes in EM datasets," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, pp. 1505–1514, 2009.

[125] S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5 –18, jan. 2004.

[126] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller, "Multiple hypothesis video segmentation from superpixel flows," in *Proceedings of the 11th European Conference on Computer Vision (ECCV 2010)*.

[127] P. Turaga, A. Veeraraghavan, and R. Chellappa, "From videos to verbs: Mining videos for activities using a cascade of dynamical systems," in *Proceedings of the 20th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007)*.

[128] Y. Pritch, A. Rav-Acha, and S. Peleg, "Nonchronological video synopsis and indexing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1971 –1984, 2008.

[129] X. Ren and J. Malik, "Tracking as repeated figure/ground segmentation," in *Proceedings of the 20th IEEE Conference on Computer Visiona nd Pattern Recognition (CVPR 2007)*.

[130] S. Liu, G. Dong, C. Yan, and S. Ong, "Video segmentation: Propagation, validation and aggregation of a preceding graph," in *Proceedings of the 21st IEEE Conference on Computer Vision and Pattern Recognition (CVPR 08)*.

[131] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[132] W. Brendel and S. Todorovic., "Video object segmentation by tracking regions," in *Proceedings of the 12th International Conference on Computer Vision (ICCV 2009)*.

[133] A. Bugeau and P. Pérez, "Track and cut: simultaneous tracking and segmentation of multiple objects with graph cuts," *Journal on Image and Video Processing*, vol. 2008, pp. 1–14.

[134] Z. Yin and R. Collins, "Shape constrained figure-ground segmentation and tracking," *Proceedings of the 21st IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*.

[135] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *Journal in ACM Computing Surveys*, vol. 38, no. 4, p. 13, 2006.

[136] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, dec 1979.

[137] A. Chan and N. Vasconcelos, "Variational layered dynamic textures," in *Proceedings of the 22nd IEEE Conference in Computer Vision and Pattern Recognition(CVPR 2009)*.

[138] V. Hedau, H. Arora, and N. Ahuja, "Matching images under unstable segmentations," in *Proceedings of the 21st IEEE Conference in Computer Vision and Patter Recognition (CVPR 2008)*.

[139] A. Ayvaci and S. Soatto, "Motion segmentation with occlusions on the superpixel graph," in *Proceedings of the 12th International Conference in Computer Vision Workshops (ICCVW 2009)*.

[140] M. Unger, T. Mauthner, T. Pock, and H. Bischof, "Tracking as segmentation of spatial-temporal volumes by anisotropic weighted TV," in *Proceedings of*

the 7th International Conference in Computer Vision and Pattern Recognition (EMMCVPR 2009).

[141] Y. Huang, Q. Liu, and D. Metaxas, "Video object segmentation by hypergraph cut," in *Proceedings of the 22nd IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*.

[142] X. Bai, J. Wang, D. Simons, and G. Sapiro, "Video snapcut: robust video object cutout using localized classifiers," *ACM Transactions in Graphics*, vol. 28, no. 3, pp. 70:1–70:11, Jul. 2009.

[143] J. Wang, Y. Xu, H.-Y. Shum, and M. F. Cohen, "Video tooning," *ACM Transactions in Graphics*, vol. 23, no. 3, pp. 574–583, Aug. 2004.

[144] D. DeMenthon, "Spatio-temporal segmentation of video by hierarchical mean shift analysis," *Statistical Methods in Video Processing Workshop 2002 (SMVP 2002)*.

[145] N. Sundaram, T. Brox, and K. Keutzer, "Dense point trajectories by gpu-accelerated large displacement optical flow," in *Proceedings of the 11th European Conference on Computer Vision (ECCV 2010)*, pp. 438–451.

[146] T.Brox and J.Malik, "Object segmentation by long term analysis of point trajectories," in *Proceedings of the 11th European Conference on Computer Vision (ECCV 2010)*.

[147] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[148] M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik, "Using contours to detect and localize junctions in natural images," in *Proceedings of the 21st International Conference on Computer Vision and Pattern Recognition (CVPR 2008)*.

[149] B. Fulkerson, A. Vedaldi, and S. Soatto, "Class segmentation and object localization with superpixel neighborhoods," in *Proceedings of the 12th International Conference in Computer Vision (ICCV 09)*.

[150] P. Kohli, M. P. Kumar, and P. H. S. Torr, "P3 & beyond: Solving energies with higher order cliques," in *20th International Conferece in Computer Vision (ICCV 2007)*.

[151] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "From contours to regions: An empirical evaluation," in *22nd IEEE Conferece in Computer Vision and Pattern Recognition (CVPR 2009)*.

[152] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., 2006.

[153] M. J. Wainwright and M. I. Jordan, *Graphical Models, Exponential Families, and Variational Inference*. Now Publishers Inc., 2008.

[154] R. Navarro, J. Arines, and R. Rivera, "Direct and inverse discrete zernike transform," *Optics Express*, vol. 17, no. 26, pp. 24 269–24 281, Dec 2009.

[155] M. Varma and A. Zisserman, "A statistical approach to material classification using image patch exemplars," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 11, pp. 2032–2047, Nov 2009.

[156] G. Zhao, T. Ahonen, J. Matas, and M. Pietikainen, "Rotation-invariant image and video description with local binary pattern features," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1465 –1477, april 2012.

[157] G. González, F. Fleuret, and P. Fua, "Learning rotational features for filament detection," in *Proceedings of the 23rd IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2009)*.

[158] A. Kreshuk, C. Straehle, C. Sommer, U. Koethe, G. Knott, and F. Hamprecht, "Automated segmentation of synapses in 3D EM data," in *IEEE International Symposium on Biomedical Imaging (ISBI 2010)*.

[159] J. Flusser, B. Zitova, and T. Suk, *Moments and Moment Invariants in Pattern Recognition*. Wiley Publishing, 2009.

[160] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. pp. 846–850, 1971.

[161] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[162] D. W. Pentico, "Assignment problems: A golden anniversary survey," *European Journal of Operational Research*, vol. 176, no. 2, pp. 774 – 793, 2007.

[163] M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *Journal of the Association for Computing Machinery*, vol. 34, no. 3, pp. 596–615, Jul. 1987.

[164] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*.

[165] M. Meila, "Comparing clusterings. An information based distance," *Journal of Multivariate Analysis*, vol. 98, no. 5, pp. 873 – 895, 2007.

[166] M. Meilă, "Comparing clusterings: an axiomatic view," in *Proceedings of the 22nd international conference on Machine learning (ICML 2005)*, pp. 577–584.

[167] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "Toward objective evaluation of image segmentation algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 929 –944, 2007.

[168] S. Kim, S. Nowozin, P. Kohli, and C. D. Yoo, "Higher-order correlation clustering for image segmentation," in *Proceedings of the 25th Conference on Neural Information Processing Systems (NIPS 2011)*.

[169] B. Liu, Y. Xia, and P. S. Yu, "Clustering through decision tree construction," in *Proceedings of the 26th international conference on management of data (SIGMOD 2000)*.

[170] H. Blockeel, L. D. Raedt, and J. Ramong, "Top-down induction of clustering trees," in *Proceedings of the 15th international conference on machine learning (ICML 1998)*.

[171] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *Journal of Artificial Intelligence Research*, vol. 4, pp. 129–145, 1996.

[172] M. Bilgic and L. Getoor, "Reflect and correct: A misclassification prediction approach to active inference," *ACM Transactions on Knowledge Discovery from Data*, vol. 3, no. 4, pp. 1–32, 2009.

[173] S. Sra, S. Nowozin, and S. J. Wright, *Optimization for Machine Learning*. The MIT Press, 2011.

[174] N. Komodakis, N. Paragios, and G. Tziritas, "MRF energy minimization and beyond via dual decomposition," *IEEE Transactions in Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 531–552, 2011.

[175] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[176] A. Billionnet, S. Elloumi, and A. Lambert, "Linear reformulations of integer quadratic programs," in *Proceedings of the 2nd international conference on*

Modeling, Computation and Optimization in Information Systems and Management Sciences (MCO 2008).